



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학박사학위논문

**Cooperative Control and Planning
for Safe Aerial Transportation in Unknown Environments**

미지 환경에서의 안전 비행 운송을 위한 협업제어 및 경로생성 기법

2017년 8월

서울대학교 대학원

기계항공공학부

이 현 범

Cooperative Control and Planning
for Safe Aerial Transportation in Unknown Environments

미지 환경에서의 안전 비행 운송을 위한 협업 제어 및 경로생성 기법

지도교수 김 현 진

이 논문을 공학박사 학위논문으로 제출함

2017년 5월

서울대학교 대학원

기계항공공학부

이 현 범

이현범의 공학박사 학위论문을 인준함

2017년 6월

위 원 장 :

김 유 단

부위원장 :

김 현 진

위 원 :

이 종 우

위 원 :

백 찬 국

위 원 :

김 우 진

Cooperative Control and Planning
for Safe Aerial Transportation in Unknown Environments

A Dissertation

by

HYEONBEOM LEE

Presented to the Faculty of the Graduate School of
Seoul National University
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical and Aerospace Engineering

Seoul National University

Supervisor : Professor H. Jin Kim

AUGUST 2017

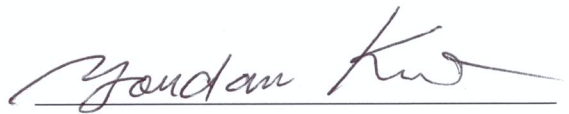
Cooperative Control and Planning
for Safe Aerial Transportation in Unknown Environments

HYEONBEOM LEE

Department of Mechanical and Aerospace Engineering

Seoul National University

APPROVED:



Youdan Kim, Chair, Ph.D.



H. Jin Kim, Ph.D.



Frank Chongwoo Park, Ph.D.



Chan Gook Park, Ph.D.



Woojin Kim, Ph.D.

to my

FATHER, MOTHER and SISTERS

with love

Abstract

Cooperative Control and Planning for Safe Aerial Transportation in Unknown Environments

Hyeonbeom Lee

Department of Mechanical and Aerospace Engineering

The Graduate School

Seoul National University

Recently, aerial manipulators using unmanned aerial vehicles (UAVs) are receiving attention due to their superior mobility in three-dimensional space. It can be applied to a wide range of applications such as inspection of hard-to-reach structure or aerial transportation. This dissertation presents a viable approach to safe aerial transportation in unknown environments by using multiple aerial manipulators. Unlike existing approaches for cooperative manipulation based on force decomposition or impedance-based control that often require heavy or expensive force/torque sensors, this dissertation suggests a method without such sensors, by exploiting the decoupled dynamics to develop estimation and control algorithms. With the decoupled dynamics and the assumption of rigid grasp, an online estimator is designed initially to estimate the mass and inertial properties of an unknown payload using the states of the aerial manipulator only. Stable adaptive controller based on the online estimated parameter is then designed using Lyapunov methods. Through simulations, the performance of the proposed controller is compared with conventional passivity-based adaptive algorithms.

This dissertation also proposes a motion generation algorithm for cooperative manipulators to transport a payload safely. If the payload is excessively heavy in comparison with the transportation ability of an aerial robot, an aerial robot may crash because of actuation limits on the motors. As a first step, the allowable flight envelope is analyzed with respect

to the position of the end-effector. In order to keep the end-effector in the allowable flight region, kinematic coordination between a payload and cooperative aerial manipulators is first studied. A two-layer framework, in which the first layer computes the motion reference of the end-effectors and the second layer calculates the joint motion of the corresponding manipulator, is then developed in a task-prioritized fashion. When generating aerial manipulator trajectories, the desired trajectory is calculated to satisfy the unilateral constraints obtained by the allowable flight envelope.

This work also considers the obstacle avoidance of cooperative aerial manipulators in unknown environments. Using the relative distance between an aerial robot and an obstacle as measured by an RGB-D camera and point cloud library (PCL), dynamic movement primitives (DMPs) modify the desired trajectory. By having the leader robot detect an obstacle and the follower robots maintain a given relative distance with the leader, improved efficiency of obstacle avoidance for cooperative robots can be achieved.

Finally, the proposed synthesis of estimation, control, and planning algorithms are validated with experiments using custom-made aerial manipulators combined with a two-DOF (Degree Of Freedom) robotic arm. The proposed method is validated with trajectory tracking using two types of payloads. Cooperative aerial transportation in unknown environments is also performed with an unknown obstacle. Both experimental results suggest that the proposed approach can be utilized for safe cooperative aerial transportation.

Keywords: Cooperative manipulation, aerial robots, obstacle avoidance, unknown payload, consensus.

Student Number: 2013-30209

Table of Contents

	Page
Abstract	vi
Table of Contents	viii
List of Tables	x
List of Figures	xi
Nomenclature	xiv
Chapter	
1 Introduction	1
1.1 Background and Motivations	1
1.2 Literature Survey	4
1.2.1 Cooperative Manipulation	4
1.2.2 Handling an Unknown Object	7
1.2.3 Obstacle Avoidance for Cooperative Robots	8
1.3 Research Objectives and Contributions	9
1.3.1 Estimation and Control Algorithm	10
1.3.2 Motion Planning within the Allowable Flight Envelope	11
1.3.3 Real-time Obstacle Avoidance using an RGB-D Camera	11
1.4 Thesis Organization	12
2 Background	14
2.1 Dynamics for Cooperative Aerial Manipulator	14
2.1.1 Rigid Body Statics	15
2.1.2 Dynamics for Single Aerial Manipulator	16
2.1.3 Decoupled Dynamics	19
2.2 Task Priority	22

2.3	DMPs	24
3	Estimator and Controller Design	26
3.1	Payload Mass and Inertia Parameter Estimation	28
3.1.1	System Parametrization	28
3.1.2	On-line Parameter Estimator	29
3.1.3	Robust Analysis for Measurement Noise	32
3.2	Controller Design	34
3.3	Simulation Results	40
4	Path Planning	45
4.1	Allowable Payload for Each Aerial Manipulator	45
4.2	Trajectory Generation with Unilateral Constraints	49
4.2.1	End-effector Trajectory Generation	49
4.2.2	Inverse Kinematics with Null Space Approach	49
4.2.3	Task Prioritization with Unilateral Constraints	56
5	Obstacle Avoidance in Unknown Environments	60
5.1	Obstacle Detection	60
5.2	Movement Primitives for Cooperative Aerial Manipulators	64
6	Experimental Validation and Results	71
6.1	Simulation Validation for Moving Obstacle	71
6.2	Experimental Setup	74
6.3	Experiment for Cooperative Aerial Transportation	77
6.3.1	Path Following with Two Types of Payloads	77
6.3.2	Aerial Transportations in Unknown Environments	78
7	Conclusions	93
	Abstract (<i>in Korean</i>)	105

List of Tables

3.1	The RMS performance for the estimation with respect to the change of the noise level	42
3.2	The RMS performance for the estimation with respect to the change of κ_1	44
3.3	Comparison results of the RMS performance	44
5.1	List of RGB-D Camera	63
6.1	The RMS errors of the ‘I’ type payload for circular trajectory tracking . . .	78
6.2	The RMS errors for the payload tracking performance	83
6.3	The RMS errors with or without rigidity maintenance algorithm	89

List of Figures

1.1	Examples of cooperative aerial manipulation and their method. (a) cooperative aerial manipulation with aerial robots and grippers [1]. (b) aerial manipulation with towed cables [2]. (c) cooperative impedance control for multiple UAVs [3]. (d) cooperative aerial transportation by optimization-based force decomposition [4]. (e) coordinated motion of aerial robotic manipulators [5].	2
1.2	An aerial manipulator delivers an object to the shelf by exploiting the robotic arm [6].	3
1.3	Examples of cooperative manipulation and their method. (a) Dual arm manipulation with impedance control [7]. (b) The leader-follower approach [8]. (c) Path planning with RRT* (Rapidly exploring Random Tree star) [9].	4
1.4	Categorization of cooperative manipulation.	6
1.5	Two aerial manipulators, each consisting of hexcopters and a robotic arm, transport an unknown common object while avoiding an unknown obstacle.	9
2.1	Coordination of two cooperative multirotors with a common payload.	15
2.2	The resultant force λ_i applied at $\Sigma_{c,i}$	19
3.1	Overall structure for the proposed synthesis.	27
3.2	Simulation environments	41
3.3	Comparison results of the parameter estimation. (Blue line: by the proposed method, Red line: by the conventional method)	43
3.4	Tracking error of the common object.(Blue line: by the proposed method, Red line: by the conventional method)	43
4.1	Configuration of an aerial manipulator.	46

4.2	Allowable payload with respect to the relative position of an object.	48
4.3	The maximum allowable payload according to the arm joint angles. η_i : the joint angle of the link i	52
4.4	Process for computing $\nu_i(j)^M$ and $\nu_i(j)^m$ (Red line : selected flight envelope, $\eta_{i,max}$: maximum joint angle of link i , and $\eta_{i,cur}$: measured joint angle of link i). . . .	53
4.5	Flight with or without return.	54
4.6	Transportation and recovery flight (①: estimation phase, ②: delivery, ③: return to base after mission failure)	55
4.7	Multiple factors for safe aerial transportation. (Task 1: trajectory for transportation, Task 2: unilateral constraints due to the allowable flight envelope, Task 3: unilateral constraints on z axis for the propeller protection.)	56
4.8	Structure for the augmented desired position of the end-effector	57
5.1	Images from RGB-D camera.	61
5.2	Outlier rejection by using voxel-grid and statistical filters. (Red circle means the outliers.)	62
5.3	Potential Function considering size of robots.	67
5.4	Virtual leader and follower structure to avoid unknown obstacles.	68
5.5	Leader-Follower relation for cooperative manipulators (No of agents: 2). . .	69
5.6	Leader-Follower relation for cooperative manipulators (No of agents: 3). . .	70
6.1	Snapshot of the simulation during 10 seconds. Two aerial manipulator systems (AMS) are used.	72
6.2	Position and attitude histories of the end-effector (a) and the object (b). . . .	73
6.3	Estimated mass.	74
6.4	Experimental setup.	75
6.5	Joule compute module [10].	76
6.6	Pictures taken from the experiment for cooperative aerial transportation ('L' shape).	79

6.7	Time histories of \mathbf{q}_o ('L' shape rod).	80
6.8	Parameter estimation ('L' shape rod).	80
6.9	Pictures taken from the experiment for cooperative aerial transportation ('I' shape).	81
6.10	Estimation and tracking performance ('I' shape rod).	82
6.11	Pictures taken from the experiment for cooperative aerial transportation.	84
6.12	Time histories of \mathbf{q}_o ('I' shape rod).	85
6.13	Parameter estimation ('I' shape rod).	85
6.14	Time histories of each end-effector with rigidity maintenance while avoiding an obstacle.	86
6.15	Time histories of each end-effector without rigidity maintenance (The proposed method).	87
6.16	Time histories of \mathbf{q}_1 and \mathbf{q}_2 without rigidity maintenance.	87
6.17	Configuration of the cooperative aerial manipulators while avoiding an obstacle.	88
6.18	Tracking errors with or without rigidity maintenance.	88
6.19	Time histories of \mathbf{q}_i and $\dot{\mathbf{q}}_i$.	90
6.20	Attitude tracking of \mathbf{q}_o and constraints for \mathbf{q}_i .	91

Nomenclature

\mathbf{I}_n : identity matrix in $\mathbb{R}^{k \times k}$
 $\mathbf{0}_k$: zero matrix in $\mathbb{R}^{k \times k}$
 l_i : length of the i -th robotic link
 $\mathbf{0}_{k \times m}$: zero vector in $\mathbb{R}^{k \times m}$
 $\mathbf{S}(\cdot)$: mapping a vector into a skew-symmetric matrix
 \dagger : the pseudo-inverse
 \mathbf{e}_3 : unit vector $\mathbf{e}_3 = [0, 0, 1]^T$
 c_* : shorthand of $\cos *$
 s_* : shorthand of $\sin *$
 $\hat{*}$: the vector or matrix which includes the estimated parameters
 $\tilde{*}_i = \hat{*} - *$: the matrix between $\hat{*}$ and $*$
 $\|*\|$: induced 2 norm of a matrix
 Σ_I : inertial frame
 Σ_b : the body frame of the hexacopter
 Σ_c : the body frame of the end effector
 $R_{b,i}$: the rotation matrix of $\Sigma_{b,i}$ w.r.t Σ_I
 R_o : the rotation matrix of Σ_o w.r.t Σ_I
 N_m : the total number of aerial manipulators
 n : DoF (Degree-of-Freedom) of the robotic arm

Defined in chapter 2

$\mathbf{p}_{b,i} = [x_{b,i}, y_{b,i}, z_{b,i}]^T$: the position of Σ_b in Σ_I
 $\Phi_i = [\phi_i, \theta_i, \psi_i]^T$: euler angles of the hexacopter
 $\eta_i = [\eta_{i,1}, \eta_{i,2}]^T$: joint angles of the manipulator
 $\mathbf{q}_i = \begin{bmatrix} \mathbf{p}_{b,i}^T & \Phi_i^T & \eta_i^T \end{bmatrix}^T$: generalized coordinate
 m_b : mass of a hexacopter

$M(\mathbf{q}_i) \in \mathbb{R}^{(6+n) \times (6+n)}$: the inertia matrix of an aerial manipulator
 $Q(\mathbf{q}_i, \dot{\mathbf{q}}_i) \in \mathbb{R}^{(6+n) \times (6+n)}$ is the Coriolis matrix of an aerial manipulator
 $W(\mathbf{q}_i) \in \mathbb{R}^{(6+n) \times 1}$: a vector for gravitational effect of an aerial manipulator
 $D(\mathbf{q}_i) \in \mathbb{R}^{(6+n) \times (6+n)}$: the inertia matrix of combined model
 $C(\mathbf{q}_i, \dot{\mathbf{q}}_i) \in \mathbb{R}^{(6+n) \times (6+n)}$: the Coriolis matrix of combined model
 $G(\mathbf{q}_i) \in \mathbb{R}^{(6+n) \times 1}$: a vector for gravitational effect of combined model
 τ_i : the control input for i -th aerial manipulator
 $\tau_{b,i}$: the control input in Σ_b
 $\mathbf{f}_i = [f_1, \dots, f_6]^T$: the input force command of the hexacopter
 $\tau_{\eta,i}$: the command for the robotic arm
 J_Φ : mapping the time derivative of Φ into angular velocity expressed in Σ_b
 M_o : motor mapping matrix
 r : arm length of the hexacopter
 k_m : the drag coefficient
 k_f : the motor thrust coefficient
 $c_m = k_m/k_f$: ratio between k_m and k_f
 ω_j : the rotational velocities of j -th rotor
 ω_j^d : the desired rotational velocities of j -th rotor
 $\lambda_i \in \mathbb{R}^{6 \times 1}$: the resulting force exerted at $\Sigma_{c,i}$
 $J_i(\mathbf{q}_i) \in \mathbb{R}^{6 \times 8}$: the Jacobian matrix from $\Sigma_{b,i}$ to $\Sigma_{c,i}$
 m_o : the mass of a payload
 J_o : the inertia of a payload
 $H_o = \text{diag}(m_o I_3, J_o)$: inertial matrix with the mass of a payload m_o and inertia J_o
 μ_o : the Coriolis matrix of a payload
 G_o : a vector for gravitational effect of a payload
 E_i : the grasp matrix
 \mathbf{r}_i : a vector between $\Sigma_{c,i}$ and Σ_o
 c_i : a constant weight for the resultant force λ_i

ω_o : angular rate of a payload

$\gamma_{i,k}$: the k -th task for the i -th aerial manipulator

$\nu_{i,k} = [\mathbf{p}_{b,i}^T, \boldsymbol{\eta}_i^T]^T$: the vector consisting of the position of $\Sigma_{b,i}$ and joint angles

$T_{i,k}$: the transformation matrix between $\dot{\nu}_{i,k}$ and $\dot{\gamma}_{i,k}$

$P = I_5 - T_{i,k}^\dagger T_{i,k}$: a projector in the null space of the transformation matrix

$\dot{\nu}_{i,k}^{\mathcal{N}}$: the homogeneous solution of $T_{i,k} P \dot{\nu}_{i,k}^{\mathcal{N}} = 0$

$T_{i,k}^A$: the augmented Jacobian matrix of the k -th task

$P_{i,k}^A$: the projector in null space of k -th task in recursive expression

$m_{i,k}$: dimension of the task

σ_j : singular value

β : the additional factor for σ_j

ϵ : the width of singular region

Defined in chapter 3

$\hat{m}_{o,i}$: estimated mass of a payload by i -th aerial manipulator

$\hat{J}_{o,i}$: estimated inertia of a payload by i -th aerial manipulator

H_* : the matrices with known physical parameters

$\mathbf{U}_i(t) = \boldsymbol{\tau}_i - M_i \ddot{\mathbf{q}}_i - Q_i \dot{\mathbf{q}}_i - W_i$: the forcing command for i -th aerial manipulator

C^* : user-defined gain matrices for a state estimator

K^* : user-defined gain matrices for a state estimator

Γ_1 : the learning rates

Γ_2 : the consensus rates

$\mathbf{e}_i = \hat{\mathbf{q}}_i - \mathbf{q}_i$: state estimation error

$\boldsymbol{\Theta}_i \in \mathbb{R}^{(6+n)}$: a bounded disturbance vector

$\rho_\theta > 0$: upper bound of $\boldsymbol{\Theta}_i$

$\mathbf{e}_{c,i} = \mathbf{q}_i - \mathbf{q}_i^d$: error between current and desired state

\mathbf{q}_i^d : desired state of \mathbf{q}_i

$\dot{\mathbf{q}}_i^r = \dot{\mathbf{q}}_i^d - \Lambda_i \mathbf{e}_{c,i}$: reference trajectory

\mathbf{s}_i : sliding surface variable

Λ_i : a diagonal gain matrix

$\boldsymbol{\xi}_i := C^* \dot{\mathbf{e}}_i + K^* \mathbf{e}_i$: a vector including state estimation error

$\boldsymbol{\xi}_i^f$: a first-order filtered value of $\boldsymbol{\xi}_i$

α_c : a time constant for $\boldsymbol{\xi}_i^f$

$\boldsymbol{\delta}_i$: the auxiliary control input

Δ_i : uncertainty of i -th aerial manipulator

K_Δ : a user-defined diagonal matrix

ρ_ξ : a upper bound value for $\|\dot{\boldsymbol{\xi}}_i\|$

ρ : a positive constant

$\boldsymbol{\chi}_i = \boldsymbol{\xi}_i^f - \boldsymbol{\xi}_i$: error between $\boldsymbol{\xi}_i^f$ and $\boldsymbol{\xi}_i$

Defined in chapter 4

$*(1, :)$: the first row vector of $*$

$\Omega_{1,\max}$: maximum speed of the rotor 1

$m_{o,i}^{\max}$: maximum allowable payload for i -th aerial manipulator

x_b^c : the length between $\Sigma_{b,i}$ and $\Sigma_{c,i}$ in x direction

$\tau_{\phi_{\max}}$: maximum roll torque command

$\tau_{\theta_{\max}}$: maximum pitch torque command

$\tau_{\psi_{\max}}$: maximum yaw torque command

$\mathbf{p}_{e,i}$: the position of the end effector in Σ_I

$\dot{\mathbf{p}}_l^d$: the augmented desired position of the end effector

$\mathbf{p}_{e,i}^d$: the desired trajectory for $\mathbf{p}_{e,i}$

$\mathbf{q}_{o,i}^d = [\mathbf{p}_o^d, \boldsymbol{\Phi}_o^d]$: the desired trajectory of the object

R_o : transforming a vector from frame Σ_o to frame Σ_I

J_η : Jacobian matrix which satisfies $\dot{\mathbf{p}}_{e,i}^b = J_\eta \dot{\boldsymbol{\eta}}_i$

$J_{e,i}$: Jacobian matrix between $\Sigma_{b,i}$ and $\Sigma_{c,i}$ without attitude effect

$B(\boldsymbol{\Phi}_i)$: a transformation matrix for the attitude between $\Sigma_{b,i}$ and $\Sigma_{c,i}$

$\boldsymbol{\nu}_i = [\mathbf{p}_{b,i}^T, \boldsymbol{\eta}_i^T]^T$: the vector consisting of the position of $\Sigma_{b,i}$ and joint angles
 $\nu_i(j)^M$: maximum joint limit of ν_i
 $\nu_i(j)^m$: minimum joint limit of ν_i
 $\boldsymbol{\gamma}_{i,1}^a := [\mathbf{p}_{e,i}; \cos(\eta_{i,1} + \eta_{i,2})]^T$: a task variable for the i -th aerial manipulator
 $\boldsymbol{\nu}_i^d$: a desired value of $\boldsymbol{\nu}_i$
 b_p : a deactivation buffer
 h_i : the smooth activation function for i -th task

Defined in chapter 5

$x_{e,i}$: the position of the end effector in x direction
 $y_{e,i}$: the position of the end effector in y direction
 $\mathbf{x}_i = [x_{e,i}, y_{e,i}]^T$: a vector for the i -th aerial manipulator
 \mathbf{o} : $x - y$ position of the unknown obstacle in Σ_I
 \mathbf{g}_i : goal position of i -th end effector
 $\zeta_{f,i}$: leader-follower relation for i -th aerial manipulator
 \mathbf{f}_i : the forcing vector for DMPs
 $\mathbf{w}_k \in \mathbb{R}^{2 \times 1}$: the weight of the k -th basis function
 α_i : the phase variable for i -th aerial manipulator
 d_{lf} : the desired distance between the leader and the follower
 r_s : the size of the boundary for the potential function
 d_a : the size of aerial manipulator
 \mathbf{d}_i : velocity command for formation control
 $\varphi(\mathbf{x}_c, \mathbf{o})$: velocity command for obstacle avoidance

1

Introduction

1.1 Background and Motivations

Simple hardware structure and affordability of multirotors have promoted the rapidly growing interests as an easy-to-work-with platform [11]. Cooperative aerial transportation as appeared in Fig. 1.1, which is one of key potential applications of multirotors, is aimed at transporting a heavier or bulkier object that cannot be handled by a single vehicle [1–5,12–16].

Aerial transportation with cable-suspended load is a well-known method to carry an object [2,12]. In these methods, however, the possible pose of a payload is limited due to the towed cables. In addition, to transport an object, human pilots should have tied the common object with cables by their own hands, so this method cannot be applied for disaster area or the place where a human cannot enter.

Another type of aerial manipulation is grasping and moving the object by using a robotic arm [17–19]. A custom-made aerial manipulator with a 2-DoF robotic arm was made in [17,18]. An aerial robot with a parallel manipulator was developed for remote safety inspection of industrial plants in [19]. However, cooperation involves complexity associated

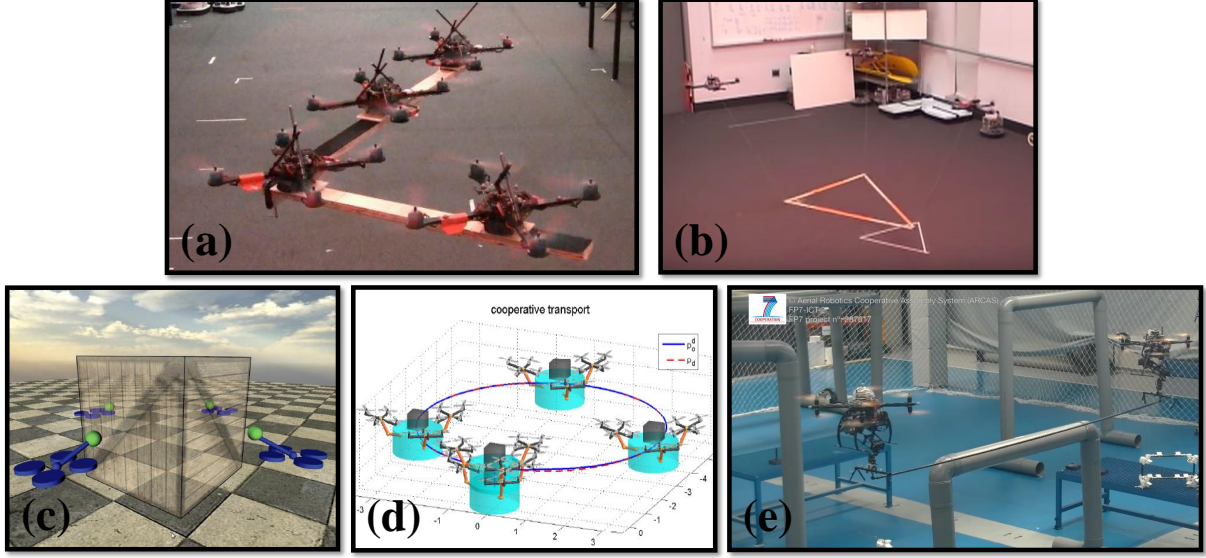


Figure 1.1: Examples of cooperative aerial manipulation and their method. (a) cooperative aerial manipulation with aerial robots and grippers [1]. (b) aerial manipulation with towed cables [2]. (c) cooperative impedance control for multiple UAVs [3]. (d) cooperative aerial transportation by optimization-based force decomposition [4]. (e) coordinated motion of aerial robotic manipulators [5].

with multiple robots in comparison with a single robot [17–19]. Many researchers have tried to solve coordination problems for multiple aerial robotic manipulators as discussed in [3–5,13]. Cooperative impedance control for multiple UAVs with a robotic arm was proposed to handle the coordination problems in [3,13]. In [4], they addressed cooperative aerial transportation by using optimization-based force decomposition method. Although these methods [3,4,13] show satisfactory simulation results, they require a multi-axis force/torque sensor to solve coordination and uncertainty problems. Force/torque sensors can be often used to estimate the unknown physical parameters of an object, but the availability of these sensors in small aerial robots could be limited because multi-axis force/torque sensors are often heavy and expensive. So, it is also impractical to assume full knowledge of physical parameters of a payload. In [5], they propose a coordinated motion algorithm for aerial robotic manipulators by only using inverse kinematics and show satisfactory experimental results. However, since this method does not handle an unknown payload, it cannot be

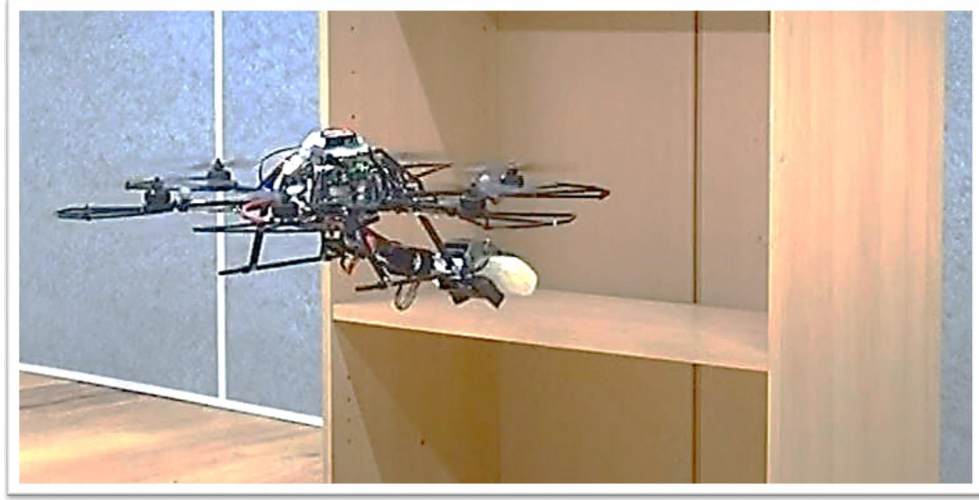


Figure 1.2: An aerial manipulator delivers an object to the shelf by exploiting the robotic arm [6].

applied for the aerial transportations in unknown environments.

Therefore, this dissertation is interested in resolving these problems: *(i)* coordination, *(ii)* uncertainty, and *(iii)* internal stability during avoiding an obstacle. First, the motion of the aerial manipulators has to be coordinated to transport an object. Second, a desired path is generated by considering actuation limits of each motor when aerial manipulators deliver an unknown object as shown in Fig. 1.2. Finally, to handle the interaction stability, aerial manipulators maintain the desired distance between aerial robots while avoiding an unknown obstacle.

Sine the aforementioned topic of cooperative aerial transportation is important and contributes to its own area, this dissertation is categorized into three detailed part: *(i)* control algorithm for cooperative manipulation, *(ii)* handling an unknown object, and *(iii)* obstacle avoidance of cooperative robots.

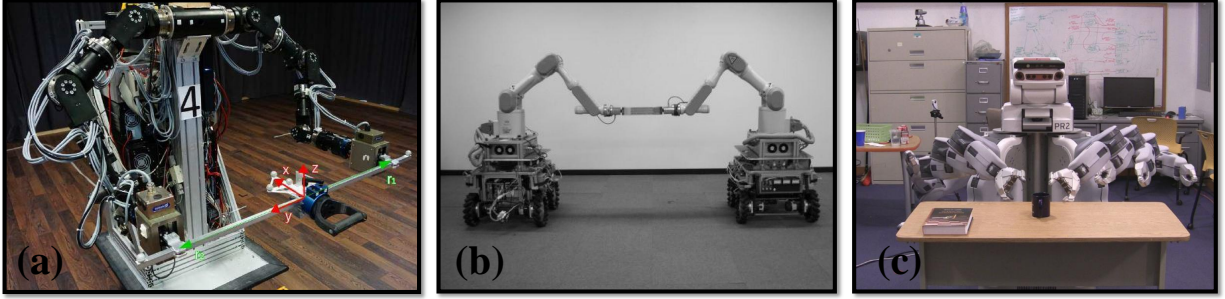


Figure 1.3: Examples of cooperative manipulation and their method. (a) Dual arm manipulation with impedance control [7]. (b) The leader-follower approach [8]. (c) Path planning with RRT* (Rapidly exploring Random Tree star) [9].

1.2 Literature Survey

This section offers the survey results of scholarly articles, book, and other sources relevant to this research.

1.2.1 Cooperative Manipulation

Cooperative manipulation can be applied to a wide range of applications in construction sites, production lines or various remote operations. Many researchers try to achieve this goal by using ground manipulators [20,21] or aerial robots [1,4,12] as shown in Fig. 1.3

These cooperative manipulations can be divided into three categories based on a task vector with respect to the target frame (or workspace in Fig. 1.4). In object level, a hybrid position and force controller based on the centralized multi-fingered dynamics is presented in [20]. A suboptimal iterative LQR-like controller based on impedance control is proposed in [21] for cooperative manipulators to carry a common object in a centralized manner. In [22,23], they present the set of feasible cable tensions for redundantly actuated cable-driven parallel robots. For multiple quadrotors with a gripper, a centralized controller is addressed in [1] to stabilize a payload along three-dimensional trajectories in a centralized manner. Unlike these fully centralized approaches, to facilitate the faster local feedback in each robot, each end-effector controller runs locally while the optimization for task planning

is performed centrally [4,12,21]. For aerial robots, in [12], they show control and planning of multiple aerial robots carrying an object by towed cables. The feasible solution for an equilibrium of a payload is numerically determined for the special case of a triangular payload. For aerial manipulators, a hierarchical control framework for multiple quadrotors with a 2-DoF robotic arm is simulated in [4]. In this method, force distribution among the end-effectors is obtained by solving constrained quadratic optimization problem. The constrained optimization [4,12] may need a relatively higher computational and communicational load.

Unlike the optimization approach, a task-oriented formulation with a kinematic coordination does not require numerical methods [24,25]. Based on the kinematic coordination, a two-layer framework, in which the first layer computes the motion reference of the end-effectors and the second layer calculates the joint motion of the corresponding manipulator, is proposed for a ground mobile manipulator with a dual arm [7]. For cooperative quadrotor manipulators, an impedance control architecture with the two-layer framework is developed to handle the contact forces at the end-effectors [13]. However, these methods [7,13] depend on multi-axis force/torque sensors to estimate and carry an unknown common object. Since the aerial robots are hard to equip heavy multi-axis force/torque sensor, they verified their algorithm only by simulation.

In order to resolve these practical difficulties associated with force/torque sensing of multiple manipulators, a coordinated motion controller based on leader-follower structure is addressed for two mobile manipulators [8]. For aerial manipulators to enable the faster attitude feedback in each aerial robot, the desired trajectories for multiple aerial manipulators are obtained by using RRT* (Rapidly-exploring Random Tree star) [14,16]. In [14], the effect of a common object was compensated based on the closed-chain dynamics in joint level. However, these methods [8,14,16] require exact knowledge of a common object.

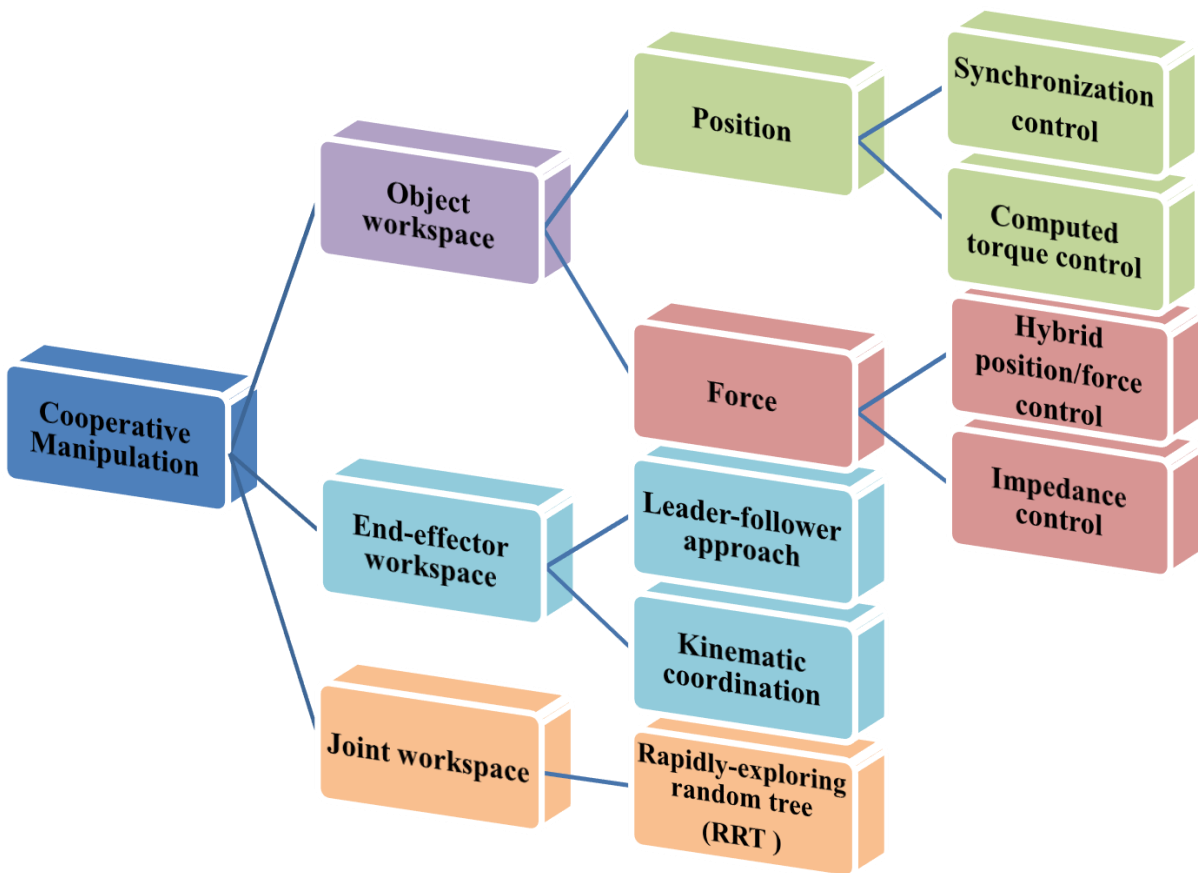


Figure 1.4: Categorization of cooperative manipulation.

1.2.2 Handling an Unknown Object

Research on handling uncertainty such as an unknown or inaccurate model has begun with early work on single manipulator [26–28]. For cooperative manipulators, there are researches to handle an inaccurate kinematic model of an object such as inaccurate orientation and length at grasp points [29,30], but they assume that physical parameters of the object such as mass or moment of inertia were known. In [31], the physical parameters of a common object and robots arms are estimated based on a distributed adaptive coordinated control method. Furthermore, consensus algorithm for an agreement on certain quantities of group interest have been studied to handle the uncertainty in [32] or synchronization of networked mobile manipulators [33]. For multiple ground mobile manipulators, in [34], a consensus for the estimation of kinematic and inertial parameters of an unknown common object is simulated by receiving the measurements of velocity and the contact force applied to the object. However, unlike these algorithms applied to ground manipulators [31,32,34], aerial manipulators need to consider the actuation limit such as speed of motors also, for stable flights.

The transportation capability of an aerial manipulator is a crucial factor in carrying a heavy load safely. In order to determine such capability of cooperative aerial manipulators, this work deals with the problems of uncertain parameters and actuation limits simultaneously, which is difficult because of the interactions between robots. Although the least-square method to estimate unknown mass attached under a multirotor [35] or robust control for parameter uncertainty of a multirotor [36] is already proposed, actuation limits of aerial robots still remain a problem. The effect of a dynamic load in a single helicopter with a gripper is considered in [37], but they do not take account of actuation limits. To avoid this issue and to operate the robots safely, in [38,39], they obtain the desired paths between a predefined initial and final configuration of a quadrotor are generated by using quadratic programming solver with constraints such as joint limits. In [40], the trajectory planning with dynamic programming is presented for a single ground mobile manipulator

to satisfy the maximum carrying capacity. However, since these algorithms solve numerical optimization problems of a single robot, complexity between multiple robots cannot be dealt with in real time.

1.2.3 Obstacle Avoidance for Cooperative Robots

For the motion planning of mobile robots, Rapidly-exploring Random Tree (RRT) is a well-known planning algorithm to generate a path in unstructured environments. Since RRT does not depend on an explicit representation of obstacles, it is popular for various applications such as obstacles avoidance [41] or motion planning of cooperative manipulators [9]. Although the fluid motion planner proposed by [42] can consider kinematic constraints such as maximum velocities, RRT can be a effective solution for high DoF robots. In [9], they experimentally validate a sampling-based planning algorithm using a dual-arm robot. They also achieve the asymptotic optimality provided by the RRT*, which is an extended version of RRT. RRT-based path planning also can be applied to aerial robots [43–45]. RRT* is used to generate the obstacle-free path for inspection operation in [43]. For aerial robots which had limited turning rates, RRT* with path smoothing method is presented based on controllable linear dynamics [44] or Bezier curve in two-dimensional plane [45]. However, since cooperative aerial manipulators have more complicate dynamics, these algorithms [44,45] are only applicable for single robots only. In addition, obstacle avoidance with RRT-based algorithms [43–45] requires a relatively heavy computer to calculate the desired path in real time, so it is hard for aerial manipulators to avoid unknown or moving obstacles in real time.

Although there exists a research to reduce the computational load [46], RRT* still needs to be improved to avoid unknown obstacles in real time. Another method for real-time obstacle avoidance, Dynamic Movement Primitives (DMPs), which represents discrete or periodic trajectories, has been exploited by re-generating smooth movements in given obstacle configuration [47–50]. For this reason, it is often used for a single robotic arm [48] to avoid a static or moving obstacle. In [49], real-time obstacle avoidance have been investi-

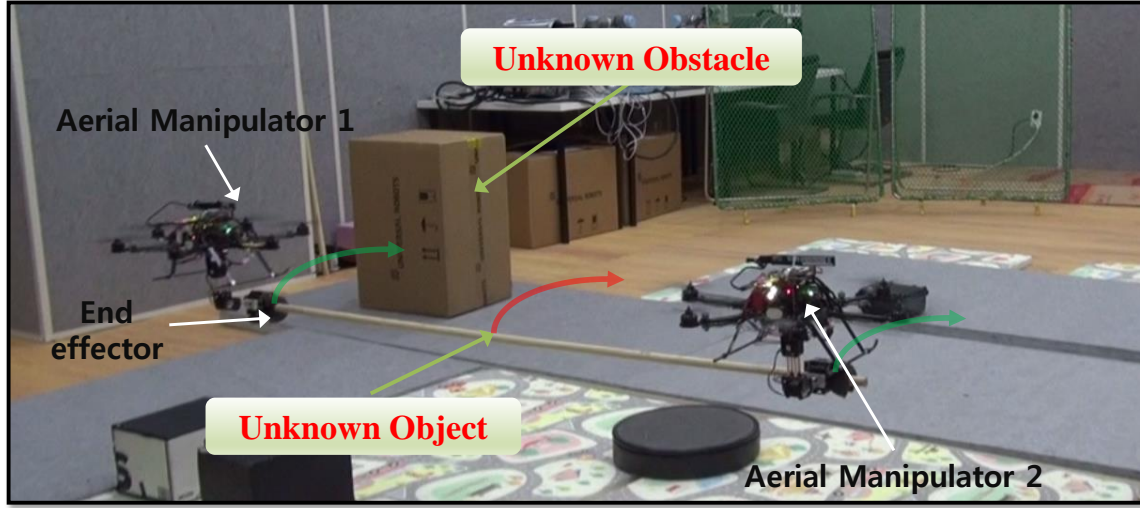


Figure 1.5: Two aerial manipulators, each consisting of hexcopters and a robotic arm, transport an unknown common object while avoiding an unknown obstacle.

gated in a bimanual task with force feedback and DMPs. In [50], cooperative manipulation and movement synchronization in the disturbance have been performed using a dual-arm manipulator. In their method, it is able to consider more than two DMPs based on formation and impedance control, but they did not consider the size of robots and an object. Since aerial robots are weak to a crash, it is more important to consider the size of robots and object unlike mobile manipulations [49,50].

1.3 Research Objectives and Contributions

This dissertation is interested in resolving the aforementioned problems for cooperative aerial manipulators in unknown environments as described in Fig. 1.5. To achieve this goal, this work concentrates on three parts: (i) estimation and control of cooperative aerial manipulators, (ii) motion planning within the allowable flight envelope, and (iii) real-time obstacle avoidance using an RGB-D camera.

1.3.1 Estimation and Control Algorithm

Objectives

The objective of the estimator and controller is to enable the aerial manipulator to handle unknown payloads without using force/torque sensors. The controller is designed by using the estimated mass to track the desired trajectory of each aerial manipulator.

Contributions

Due to the interaction with a common object, the dynamics of cooperative aerial manipulators is more complicated than that of a single robot or cable-suspended robots. For this reason, many researchers designed the controller for cooperative manipulators based on centralized approach. However, the attitude stabilizer of an aerial robot should run fast (over 100 Hz), which is out of measurement update rate in the position. Therefore, the centralized approach for aerial manipulators is not recommendable. To resolve this problem, this work derives the decoupled dynamics of cooperative aerial manipulators. Based on the rigid grasp assumption, the decoupled dynamics can be represented not by the states of a payload but by the states of the aerial manipulator only.

The decoupled dynamics in the joint space facilitate faster attitude stabilizer and state estimation of the aerial manipulator. Based on state estimation in decoupled dynamics, this work proposes an on-line parameter estimator for handling an unknown payload. Since the state estimators run in a decentralized manner, the estimated parameter can differ from each other. To handle this issue, following the preliminary research of this work [6,51], cooperative aerial manipulators share the estimated physical parameters of the common object. Utilization of the consensus algorithm with such information exchange improves the estimation performance. Although decoupled dynamics makes the controller run on the joint space, the stability issues remain a problem. By using Lyapunov convergence analysis, this work proves that the controller with estimated parameter is stable.

1.3.2 Motion Planning within the Allowable Flight Envelope

Objectives

Second, the planning algorithm is proposed to efficiently generate the desired path of each aerial manipulator. By using the planning algorithm, the end-effector should remain in the allowable flight region where the aerial manipulators can avoid the excess of the actuation limits.

Contributions

Unlike ground manipulators, aerial manipulators are prone to crash because of actuation limits of each motor. So, when carrying a heavy payload, the safety should be considered. To handle this issue, this dissertation proposes a safe motion generation algorithms depending on the physical properties of an unknown payload such as unknown mass. The desired trajectory for each aerial manipulator is generated by a two-layer framework for the motion generation, in which the first layer computes the desired trajectory of the end-effectors and the second layer calculates the joint motion of the corresponding manipulator as same with [7]. However, unlike the research in [7], this work considers the actuation limits when calculating the desired path of aerial manipulators. By exploiting the proposed on-line parameter estimator, the desired trajectory of the aerial manipulator is adjusted automatically based on the estimated parameters.

In order to adjust the desired path in real time, the allowable flight envelope of aerial manipulator is investigated. Considering the maximum speed of each motor and the configuration of the robotic arm, the flight envelope was given with respect to the location of the end-effector. By using the flight envelope, this work does not depend on any numerical optimization, so the computational loads can decrease.

1.3.3 Real-time Obstacle Avoidance using an RGB-D Camera

Objectives

Third, cooperative aerial manipulators can also avoid unknown obstacles such as other

drones or buildings by exploiting DMPs. The more detailed research objectives and contributions of each part are described as follows.

Contributions

If one aerial robot avoid an obstacle and other robots want to follow its own desired trajectory without modifying the trajectory, the internal force between each other may increase. Since the internal force can interrupt the stability of the whole system, so is important to solve the aforementioned issue. For this reason, obstacle avoidance of the cooperative robots is more complicated than a single robot. To resolve this issue, this dissertation proposes obstacle avoidance algorithms for cooperative aerial manipulators by using DMPs and RGB-D camera.

By exploiting DMPs, aerial manipulators calculate the safety boundary based on the size of robots and modify the desired trajectories of each end-effector. Since DMPs does not change the target position while avoiding unknown obstacles, the final position is guaranteed to be preserved. In the aspect of efficiency, virtual leader-follower structure is employed while avoiding an unknown obstacle. This structure can reduce the communication load when multiple aerial manipulators need to avoid flying obstacles. In order to detect an unknown obstacle, this work uses RGB-D camera based on point cloud library [52] without assuming the known position of an obstacle. By using a voxel-grid filter in PCL [53], outliers of depth data are efficiently removed. Thanks to an onboard camera and DMPs, the proposed avoidance algorithm can be easily adapted for various cooperative robots, not limited to cooperative aerial manipulators.

1.4 Thesis Organization

The rest of this dissertation is organized as follows. Chapter 2 describes the necessary materials for this dissertation including decoupled dynamics of cooperative aerial manipulators, task priority, and DMPs. Chapter 3 proposes on-line parameter estimator and the controller for handling an unknown payload. A motion generator to carry a common object and the

algorithm for obstacle avoidance are shown in chapter 4 and 5. Chapter 6 shows the experimental results using custom-made aerial manipulator to verify our proposed synthesis in real environments. Chapter 7 summarizes the considered issues and concluding remarks in this dissertation.

2

Background

In this section, the necessary backgrounds were presented including dynamics of aerial manipulators, task priority and DMPs (Dynamic Movement Primitives).

2.1 Dynamics for Cooperative Aerial Manipulator

This work considers that each aerial manipulator consists of a hexacopter and a 2-DoF arm. The coordinated frames $\Sigma_I, \Sigma_b, \Sigma_c$ represent the inertial frame, the body frame of the hexacopter and the body frame of the end-effector, respectively as shown in Fig. 2.1. Σ_o means the body frame of the object.

For the i -th manipulator, using the position of center of mass of the hexacopter in the inertial frame $\mathbf{p}_{b,i} = [x_{b,i}, y_{b,i}, z_{b,i}]^T$, Euler angles of the hexacopter $\Phi_i = [\phi_i, \theta_i, \psi_i]^T$ and joint angles of the manipulator $\boldsymbol{\eta}_i = [\eta_{i,1}, \eta_{i,2}]^T$.

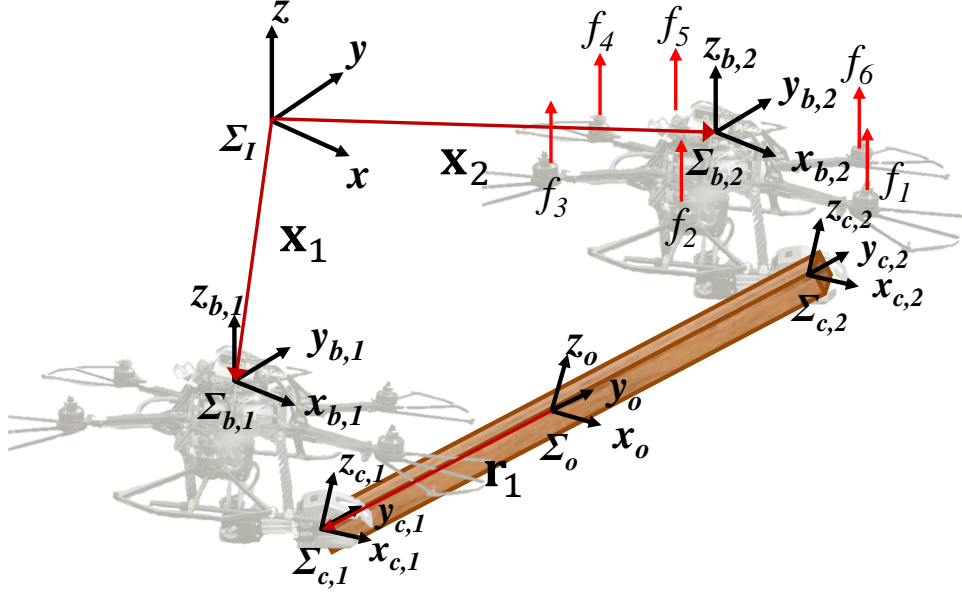


Figure 2.1: Coordination of two cooperative multirotors with a common payload.

2.1.1 Rigid Body Statics

In the following, this work refers the general case in which each aerial manipulator can apply forces and torques to the common payload. The force and torque applied by the i -th end-effector are concatenated to the *wrench* denoted $\lambda_i \in \mathbb{R}^6$. The effective wrench acting on the origin of the common payload Σ_o is denoted λ_o and is determined by the end-effector wrench as

$$\lambda_o = \sum_{i=1}^{N_m} E_i \lambda_i. \quad (2.1)$$

Here, N_m is the total number of aerial manipulators and E_i is the grasp matrix [54] which depends on the kinematic parameters \mathbf{r}_i as

$$E_i = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ S(\mathbf{r}_i) & \mathbf{I}_3 \end{bmatrix}. \quad (2.2)$$

$S(*)$ denotes the skew symmetric matrix.

In addition, since this work considers the three-dimensional space, basic materials such as $se(3)$ and $twist$ are provided based on the formulation described in [55]. First, for a matrix $R \in \mathbb{R}^{3 \times 3}$ or a vector $\omega \in \mathbb{R}^3$, the definition of $SO(3)$ and $so(3)$ is given as

$$\begin{aligned} SO(3) &= \{R \in \mathbb{R}^{3 \times 3} : RR^T = I_{3 \times 3}, \det R = +1\} \\ so(3) &= \{S(\omega) \in \mathbb{R}^{3 \times 3} : S(\omega)^T = -S(\omega)\}, \end{aligned} \quad (2.3)$$

where $I_{3 \times 3}$ is the 3×3 identity matrix. The notation of SO abbreviates *special, orthogonal*. Note that given a skew-symmetric matrix $S(\omega) \in so(3)$ and $\theta \in \mathbb{R}$, $\exp(S(\omega)\theta) \in SO(3)$. Also, it is well known that any rotation matrix $R \in SO(3)$ is equivalent to a rotation about a fixed axis $\omega \in \mathbb{R}^3$ through an angle $\theta \in (0, 2\pi]$.

From the definition of $SO(3)$ and $so(3)$, special Euclidean group $SE(3)$ and $se(3)$ are defined as

$$\begin{aligned} SE(3) &= \{(p, R) : p \in \mathbb{R}^3, R \in SO(3)\} = \mathbb{R}^3 \times SO(3) \\ se(3) &= \{(v, S(\omega)) : v \in \mathbb{R}^3, S(\omega) \in so(3)\} = \mathbb{R}^3 \times so(3). \end{aligned} \quad (2.4)$$

Here, in general, v means the velocity of a joint, i.e., $\dot{p}(t) = v$.

An element of $se(3)$ is referred to as a *twist*, or a (infinitesimal) generator of the Euclidean group. So, $\xi := (v, S(\omega))$ is called *twist coordinates* of $S(\xi)$. Note that given $S(\xi) \in se(3)$ and $\theta \in \mathbb{R}$, $\exp(S(\xi)\theta) \in SE(3)$.

2.1.2 Dynamics for Single Aerial Manipulator

The kinematic model can be described based on the following system state,

$$\mathbf{q}_i = \begin{bmatrix} \mathbf{p}_{b,i}^T & \Phi_i^T & \boldsymbol{\eta}_i^T \end{bmatrix}^T, \quad (2.5)$$

for $i = 1, \dots, N_m$. Although we use two aerial manipulators in experiments, the proposed algorithm can be easily extended to multiple aerial manipulators. In general, we will use bold letters (e.g., \mathbf{q}_i , $\mathbf{\Phi}_i$) to indicate vector quantities.

The Euler-Lagrange formulations for obtaining the dynamics of the combined system can be written as

$$\frac{d}{dt} \frac{\delta L}{\delta \dot{\mathbf{q}}_i} - \frac{\delta L}{\delta \mathbf{q}_i} = \boldsymbol{\tau}_i, \quad (2.6)$$

where the control input $\boldsymbol{\tau}_i$ consists of six-dimensional force/torque of the multirotor and two joint torque components of the arm. Here, $L = K(\mathbf{q}_i, \dot{\mathbf{q}}_i) - P(\mathbf{q}_i)$ with kinetic energy $K = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}_i) \dot{\mathbf{q}}$ and potential energy $P(\mathbf{q}_i)$. With the mass of hexacopter m_b and the total mass of the robotic arm, following the results of [56], the dynamics of the combined system can be written as

$$M(\mathbf{q}_i) \ddot{\mathbf{q}}_i + Q(\mathbf{q}_i, \dot{\mathbf{q}}_i) \dot{\mathbf{q}}_i + W(\mathbf{q}_i) = \boldsymbol{\tau}_i, \quad (2.7)$$

where $M(\mathbf{q}_i) \in \mathbb{R}^{(6+n) \times (6+n)}$ is the inertia matrix, $Q(\mathbf{q}_i, \dot{\mathbf{q}}_i) \in \mathbb{R}^{(6+n) \times (6+n)}$ is the Coriolis matrix, $W(\mathbf{q}_i) \in \mathbb{R}^{(6+n) \times 1}$ is the gravity term and n is DoF (Degree-of-Freedom) of the robotic arm.

In the case of the multirotor as shown in Fig. 2.1, the control input $\boldsymbol{\tau}_i$ can be converted into the actuation command as

$$\boldsymbol{\tau}_i = R_c \boldsymbol{\tau}_{b,i} = R_c N_c \mathbf{h}_i := \Xi \mathbf{h}_i \quad (2.8)$$

where $\boldsymbol{\tau}_{b,i}$ is the control input in the body frame Σ_b , $\mathbf{h}_i = [\mathbf{f}_i^T, \boldsymbol{\tau}_{\eta,i}^T]^T$ consists of $\mathbf{f}_i = [f_1, \dots, f_6]^T$ for the input force command of the hexacopter and $\boldsymbol{\tau}_{\eta,i}$ for the command to the arm. $R_c = \text{diag}(R_{b,i}, J_\Phi^T, I_n)$ is $\mathbb{R}^{(6+n) \times (6+n)}$ matrix including the rotation matrix $R_{b,i}$ of $\Sigma_{b,i}$ with respect to Σ_I , $n \times n$ identity matrix I_n and the matrix J_Φ which maps the time derivative of $\mathbf{\Phi}$ into angular velocity expressed in Σ_b . The $\mathbb{R}^{(6+n) \times (6+n)}$ matrix $N_c = \text{diag}(M_o, I_n)$

includes the matrix M_o which can be defined as

$$M_o = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ s_{30^\circ} r & r & s_{30^\circ} r & -s_{30^\circ} r & -r & -s_{30^\circ} r \\ -c_{30^\circ} r & 0 & c_{30^\circ} r & c_{30^\circ} r & 0 & -c_{30^\circ} r \\ -c_m & c_m & -c_m & c_m & -c_m & c_m \end{bmatrix},$$

where r is arm length of the hexacopter and $c_m = k_m/k_f$ with k_m the drag coefficient and k_f the motor thrust coefficient. The coefficients k_f and k_m are obtained from motor test data using a six-axis force-torque sensor. Here, since $\Xi^T \Xi$ is always invertible except when $\theta = \pm k \frac{\pi}{2}$ with $k = 1, 3, 5, \dots$ [57], we can obtain the desired thrust for each motor and the robotic arm as

$$\mathbf{h}_i = \Xi^\dagger \boldsymbol{\tau}_i, \quad (2.9)$$

where † is the Moore-Penrose pseudo-inverse. In this case, since $f_j = k_f(\omega_j^d)^2$ with the rotational velocities of j -th rotor ω_j^d , we can give the speed command for each motor as $\omega_j^d = \sqrt{f_j/k_f}$ [58].

From (2.9), we can consider the control input as $\boldsymbol{\tau}_i$ only in the remainder of the paper. In addition, to satisfy the relationship in (2.9), two elements of $\boldsymbol{\tau}_i$, i.e. $\boldsymbol{\tau}_i(1)$ and $\boldsymbol{\tau}_i(2)$, are used to generate the desired roll ϕ^d and pitch angle θ^d . From (2.8), we can obtain $\boldsymbol{\tau}_i(1 : 3) = R[0, 0, \sum_{j=1}^6 h_j]^T$ for the control input in x , y and z direction. Using the fact that the sum of the forces generated by individual motors is the same as the altitude control input, i.e., $\sum_{j=1}^6 h_j = \boldsymbol{\tau}_{b,i}(3)$, the following equation should be satisfied:

$$\begin{bmatrix} \boldsymbol{\tau}_i(1) \\ \boldsymbol{\tau}_i(2) \end{bmatrix} = \boldsymbol{\tau}_{b,i}(3) \begin{bmatrix} c_{\phi^d} s_{\theta^d} c_{\psi^d} + s_{\phi^d} s_{\psi^d} \\ c_{\phi^d} s_{\theta^d} s_{\psi^d} - s_{\phi^d} c_{\psi^d} \end{bmatrix}, \quad (2.10)$$

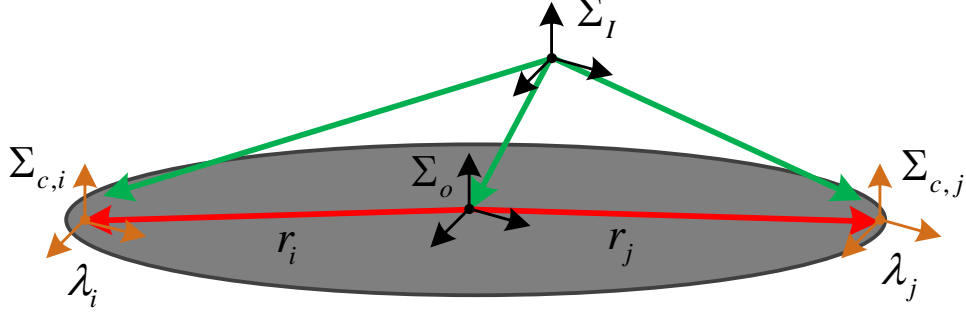


Figure 2.2: The resultant force λ_i applied at $\Sigma_{c,i}$.

where c_* is $\cos(*)$, s_* is $\sin(*)$ and the superscript d means the desired value (See [59] more detail). For this reason, the desired roll and pitch angles are computed as

$$\begin{bmatrix} \theta^d \\ \phi^d \end{bmatrix} = \frac{1}{\tau_{b,i}(3)} \begin{bmatrix} c_{\psi^d} & s_{\psi^d} \\ s_{\psi^d} & -c_{\psi^d} \end{bmatrix} \begin{bmatrix} \tau_i(1) \\ \tau_i(2) \end{bmatrix}. \quad (2.11)$$

This derivation is derived based on the small angle assumption in roll and pitch angles, i.e., $c_* \approx 1$ and $s_* \approx *$.

2.1.3 Decoupled Dynamics

When the aerial manipulator and the object interact, the resulting force $\lambda_i \in \mathbb{R}^{6 \times 1}$ is exerted at the end-effector in $\Sigma_{c,i}$ of the i -th aerial manipulator (See Fig. 2.2). In this case, the dynamics of aerial manipulator is different from the single aerial manipulator alone. Considering the resulting force λ_i and the state \mathbf{q}_i , the equation of motion of the i -th the aerial manipulator can be represented as

$$M_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + Q_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i + W_i(\mathbf{q}_i) = \tau_i - J_i^T(\mathbf{q}_i)\lambda_i, \quad (2.12)$$

where $J_i(\mathbf{q}_i) \in \mathbb{R}^{6 \times 8}$ means the Jacobian matrix from $\Sigma_{b,i}$ to $\Sigma_{c,i}$.

To obtain the dynamics of the rigid object, we define the twist $\dot{\mathbf{q}}_o = [\dot{\mathbf{p}}_o^T, \boldsymbol{\omega}_o^T]^T \in se(3)$ as a six-dimensional vector composed of translational velocity $\dot{\mathbf{p}}_o \in \mathbb{R}^3$ and rotational velocity

of the object $\boldsymbol{\omega}_o \in \mathbb{R}^3$ in Σ_I .

Using $H_o = \text{diag}(m_o I_3, J_o)$ with the mass of object m_o and inertia J_o , the dynamics of rigid object can be written:

$$H_o \ddot{\mathbf{q}}_o + \mu_o \dot{\mathbf{q}}_o + G_o = \boldsymbol{\lambda}_o, \quad (2.13)$$

where μ_o and G_o can be written as

$$\mu_o = \begin{bmatrix} 0_3 & 0_3 \\ 0_3 & S(\boldsymbol{\omega}_o) J_o \end{bmatrix}, \quad G_o = \begin{bmatrix} -m_o g \mathbf{e}_3 \\ \mathbf{0}_{3 \times 1} \end{bmatrix}.$$

Here, $\mathbf{e}_3 = [0, 0, 1]^T$, I_3 and 0_3 are 3×3 identity and zero matrices, respectively.

In our configuration, with the assumption of rigid grasp, all positions and orientations of the common object and the end-effectors coordinates can be expressed relative to a common reference frame. In this case, we can obtain the relationship between \mathbf{q}_i and \mathbf{q}_o as

$$\dot{\mathbf{q}}_o = E_i^{-T} J_i \dot{\mathbf{q}}_i. \quad (2.14)$$

Using eq. (2.14), we can combine the dynamics of the i -th aerial manipulator and the object with the force distribution solution [60]. Then we obtain the resultant force $\boldsymbol{\lambda}_i$ as

$$\boldsymbol{\lambda}_i = c_i E_i^\dagger (H_o \ddot{\mathbf{q}}_o + \mu_o \dot{\mathbf{q}}_o + G_o), \quad (2.15)$$

where E_i^\dagger can be obtained by the Moore-Penrose pseudo-inverse solution [60] as

$$c_i E_i^\dagger = \begin{bmatrix} c_i I_3 & -c_i S(\mathbf{r}_i) \Pi^{-1} \\ 0_3 & c_i \Pi^{-1} \end{bmatrix}, \quad (2.16)$$

where $\Pi = I_3 + \sum_{i=1}^N c_i S(\mathbf{r}_i) S^T(\mathbf{r}_i)$. Here, if $E^\dagger = [c_1 E_1^\dagger, \dots, c_N E_N^\dagger]^T$ and $E = [E_1, \dots, E_N]^T$, it also satisfies $EE^\dagger = I$. Note that a simpler way of load distribution solution can be found

in [61]. However, the torque solution in this method shows a larger value than the solution of the method in [60] (see the more detail in [60,61]). Since large torques cannot be well maintained by the multirotors, we used different distribution solution to reduce amount of torque. We can use unequal values for c_i as long as $\sum_{i=1}^N c_i = 1$. However, for simplicity, we assume that all torques at the center of the object in Σ_o are equally distributed among the aerial manipulators, so we set $c_i = 1/N$.

To compute \mathbf{r}_i , we need the geometric configuration of the end-effector and the object. In order not to use force/torque sensors, we use the assumption of uniform mass distribution, so that the geometric centroid coincides with the center of mass of the object.

Assumption 1. *The payload object is rigid with uniform mass distribution.*

Then, the equation of motion of the i -th aerial manipulator with the object can be rewritten as

$$D_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i + G_i(\mathbf{q}_i) = \boldsymbol{\tau}_i. \quad (2.17)$$

Here, the matrices are computed as

$$\begin{aligned} D_i &= M_i(\mathbf{q}_i) + c_i M_o(\mathbf{q}_i) \\ C_i &= Q_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + c_i Q_o(\mathbf{q}_i, \dot{\mathbf{q}}_i) + c_i J_i^T (E_i^\dagger H_o E_i^{-T}) \dot{J}_i \\ G_i &= W_i + c_i W_o(\mathbf{q}_i) \end{aligned} \quad (2.18)$$

where the following representation can be taken from the results in the preliminary work of this research [14]

$$\begin{aligned} M_o(\mathbf{q}_i) &= J_i^T (E_i^\dagger H_o E_i^{-T}) J_i \\ Q_o(\mathbf{q}_i, \dot{\mathbf{q}}_i) &= J_i^T (E_i^\dagger \mu_o E_i^{-T}) \dot{J}_i \\ W_o(\mathbf{q}_i) &= J_i^T E_i^\dagger G_o. \end{aligned} \quad (2.19)$$

E_i^\dagger can be obtained by the Moore-Penrose pseudo-inverse in [60] as

$$c_i E_i^\dagger = \begin{bmatrix} c_i I_3 & -c_i S(\mathbf{r}_i) \Pi^{-1} \\ 0_3 & c_i \Pi^{-1} \end{bmatrix}, \quad (2.20)$$

where c_i is a constant weight such that $\sum_{i=1}^{N_m} c_i = 1$ and $\Pi = I_3 + \sum_{i=1}^{N_m} c_i S(\mathbf{r}_i) S^T(\mathbf{r}_i)$.

Remark 1. *With any arbitrary vector $\mathbf{s} \in \mathbb{R}^{8 \times 1}$, we observe*

$$\begin{aligned} \mathbf{s}^T (\dot{D}_i - 2C_i) \mathbf{s} &= \mathbf{s}^T [\dot{M}_i - 2Q_i + c_i (\dot{M}_o - 2Q_o) - 2c_i (J_i^T (E_i^\dagger H_o E_i^{-T}) \dot{J}_i)] \mathbf{s} \\ &= c_i \mathbf{s}^T [J_i^T E_i^\dagger (\dot{H}_o - 2\mu) E_i^{-T} J_i] \mathbf{s}. \end{aligned} \quad (2.21)$$

In this case, applying the fact that $\dot{J}_o = S(\boldsymbol{\omega}_o) J_o - J_o S(\boldsymbol{\omega}_o)$, $\dot{H}_o - 2\mu_o$ can be rewritten as

$$(\dot{H}_o - 2\mu)^T = \begin{bmatrix} 0_3 & 0_3 \\ 0_3 & -S(\boldsymbol{\omega}_o) J_o - J_o S(\boldsymbol{\omega}_o) \end{bmatrix}. \quad (2.22)$$

Since $-S(\boldsymbol{\omega}_o) J_o - J_o S(\boldsymbol{\omega}_o)$ is a skew symmetric matrix, so is $(\dot{H}_o - 2\mu_o)^T$ in (2.22) (See [62] for more detail). However, because $E_i^\dagger \neq E_i^{-1}$, the term $\dot{D}_i - 2C_i$ may not satisfy the skew-symmetric property. Nevertheless, the term $-S(\boldsymbol{\omega}_o) J_o - J_o S(\boldsymbol{\omega}_o)$ in (2.22) can be easily computed using the angular velocity of the end-effector, which is the same as $\boldsymbol{\omega}_0$ due to the rigid grasp arm assumption.

2.2 Task Priority

We employ the formulation of task priority [63,64] to generate trajectories satisfying unilateral constraints or maintaining the safety envelope of the end-effector, which are referred to as tasks. The description in this section is based on [63,64], and the details can be found therein.

Consider the k -th task for the i -th aerial manipulator with the differential kinematic

equation:

$$\dot{\boldsymbol{\gamma}}_{i,k} = T_{i,k} \dot{\boldsymbol{\nu}}_{i,k}, \quad i = 1, \dots, N_m \quad (2.23)$$

where $\boldsymbol{\gamma}_{i,k} \in \mathbb{R}^{m_{i,k}}$ is the task vector representing the Cartesian coordinate of the end-effector for the task k and $\boldsymbol{\nu}_{i,k} = [\mathbf{p}_{b,i}^T, \boldsymbol{\eta}_i^T]^T \in \mathbb{R}^5$ is the vector consisting of the position and joint angles of the aerial manipulator. $T_{i,k}$ is the transformation matrix between $\dot{\boldsymbol{\nu}}_{i,k}$ and $\dot{\boldsymbol{\gamma}}_{i,k}$. Note that the dimension of the vector $\dot{\boldsymbol{\gamma}}_{i,k}$ can vary depending on the specific task defined by user.

In order to obtain the trajectory in the space of $\boldsymbol{\nu}_{*,*}$, we can use the following equation as

$$\dot{\boldsymbol{\nu}}_{i,k} = T_{i,k}^\dagger \dot{\boldsymbol{\gamma}}_{i,k} + P \dot{\boldsymbol{\nu}}_{i,k}^\mathcal{N} = \dot{\boldsymbol{\nu}}_{i,k}^\mathcal{N} + T_{i,k}^\dagger (\dot{\boldsymbol{\gamma}}_{i,k} - T_{i,k} \dot{\boldsymbol{\nu}}_{i,k}^\mathcal{N}) \quad (2.24)$$

where \dagger is the pseudo-inverse, $P = I_5 - T_{i,k}^\dagger T_{i,k}$ is a projector in the null space of the transformation matrix and I_n is 5×5 identity matrix. $\dot{\boldsymbol{\nu}}_{i,k}^\mathcal{N}$ is the homogeneous solution of $T_{i,k} P \dot{\boldsymbol{\nu}}_{i,k}^\mathcal{N} = 0$. In general, singular value decomposition (SVD) is used to calculate the pseudo-inverse of $T_{i,k}$. In this case, (2.24) can be computed when the Jacobian matrix $T_{i,k}$ is full rank. However, the pseudo-inverse of the Jacobian matrix may not exist at singularities or in their neighbourhood. To resolve this issue, we use Jacobian Damping (JD) as described in [65]. For the singular value of the Jacobian matrix $T_{i,k}$, i.e., $\sigma_j \geq 0 (j = 1, \dots, m_{i,k})$, the singular value can be approximated as

$$\frac{1}{\sigma_j} \approx \frac{\sigma_j}{\sigma_j^2 + \beta^2}. \quad (2.25)$$

Here, the factor β can be computed as

$$\beta^2 = \begin{cases} 0, & \text{if } \sigma_{\min} \geq \epsilon \\ (1 - (\sigma_{\min}/\epsilon)^2) \beta_{\max}^2, & \text{otherwise} \end{cases}, \quad (2.26)$$

where the parameter $\epsilon > 0$ the width of singular region, σ_{\min} is the smallest singular value and β_{\max} is the maximum damping factor allowed.

In the standard task priority framework [63], the k -th task is performed along the direction of ‘not disturbing’ the $k-1$ -th higher priority task. The hierarchy of multiple tasks is computed by projecting the k -th task in the null space of all the higher priority tasks as

$$\dot{\boldsymbol{\nu}}_{i,k} = \dot{\boldsymbol{\nu}}_{i,k-1} + (T_{i,k}P_{i,k-1}^A)^\dagger(\dot{\boldsymbol{\gamma}}_{i,k} - T_{i,k}\dot{\boldsymbol{\nu}}_{i,k-1}), \quad (2.27)$$

initialized with the zero matrix i.e., $\dot{\boldsymbol{\nu}}_{i,1} = 0_{5 \times 1}$. $P_{i,k}^A$ is the projector in the null space of the augmented Jacobian matrix of the k -th task as

$$T_{i,k}^A = \begin{bmatrix} T_{i,1} & \cdots & T_{i,k} \end{bmatrix}^T. \quad (2.28)$$

Note that the projector in null space of k -th task can be calculated by using recursive expression [64] as

$$P_{i,k}^A = P_{i,k-1}^A - (T_{i,k}P_{i,k-1}^A)^\dagger T_{i,k}P_{i,k-1}^A, \quad (2.29)$$

initialized with the identity matrix i.e., $P_{i,0} = I_n$.

2.3 DMPs

DMPs can represent complex movements with incorporating sensory feedback in real time [47]. Details about DMPs are described in [66]. Here, basic information for DMPs are addressed only based on the formulation used in [48].

DMPs can be defined based on the following attractor dynamics as

$$\begin{aligned} \dot{v}_t &= K_p(g_t - x_t) - B_tv_t + K_tf_t(\alpha) \\ \dot{x}_t &= v_t, \end{aligned} \quad (2.30)$$

where x_t and v_t are trajectory and velocity for one DoF of the system. g_t means the goal of attractor dynamics. K_t corresponds to the spring constant and B_t is the damping gain. Subscript t means the state by DMPs. Here, $f_t(\alpha)$ is a forcing term to describe the nonlinear motion as

$$f_t(\alpha) = \frac{\sum_{i=1}^{N_w} \omega_i \Psi_i(\alpha)}{\sum_{i=1}^{N_w} \Psi_i(\alpha)} \alpha, \quad (2.31)$$

with ω_i is the weight of each basis function. The exponential basis function $\Psi_i(\alpha)$ ($i = 1, \dots, N_w$) can be defined as

$$\Psi_i(\alpha) = \exp\left(-\frac{(\alpha - b_i)^2}{2\sigma_i^2}\right),$$

where b_i and σ_i are constants that determine the width and center of the basis function, respectively.

As shown in (2.30), $f_t(\alpha)$ does not depend on time. Instead it depends on a phase variable α , which varies from 1 to 0 during a movement. The phase variable α is defined as

$$\dot{\alpha} = -\gamma_t \alpha, \quad \gamma_t > 0, \quad (2.32)$$

where γ_t is a predefined gain.

3

Estimator and Controller Design

In this section, on-line parameter estimator for an unknown payload is designed. Based on the estimated parameter, the controller for each aerial manipulator is proposed. The total control structure is described in Fig. 3.1. The reference trajectories of the end-effectors are generated by kinematic coordination. In this case, since this work assumed that the geometry of the payload is known, the trajectory can be calculated in on-line framework or even in off-line framework. The trajectory of the each end-effector is learned by DMPs, then the trajectory is regenerated in onboard computer on the aerial manipulator. Then, each aerial manipulator follows its own trajectory calculated by the trajectory of the corresponding end-effector, while estimating the parameter of an unknown payload in real time and avoiding an unknown obstacle. To improve the estimation performance, the estimation result is communicated to other aerial manipulator(s). One of the biggest advantages of this structure is that the controller does not require any other measurement except the states of its own aerial manipulator.

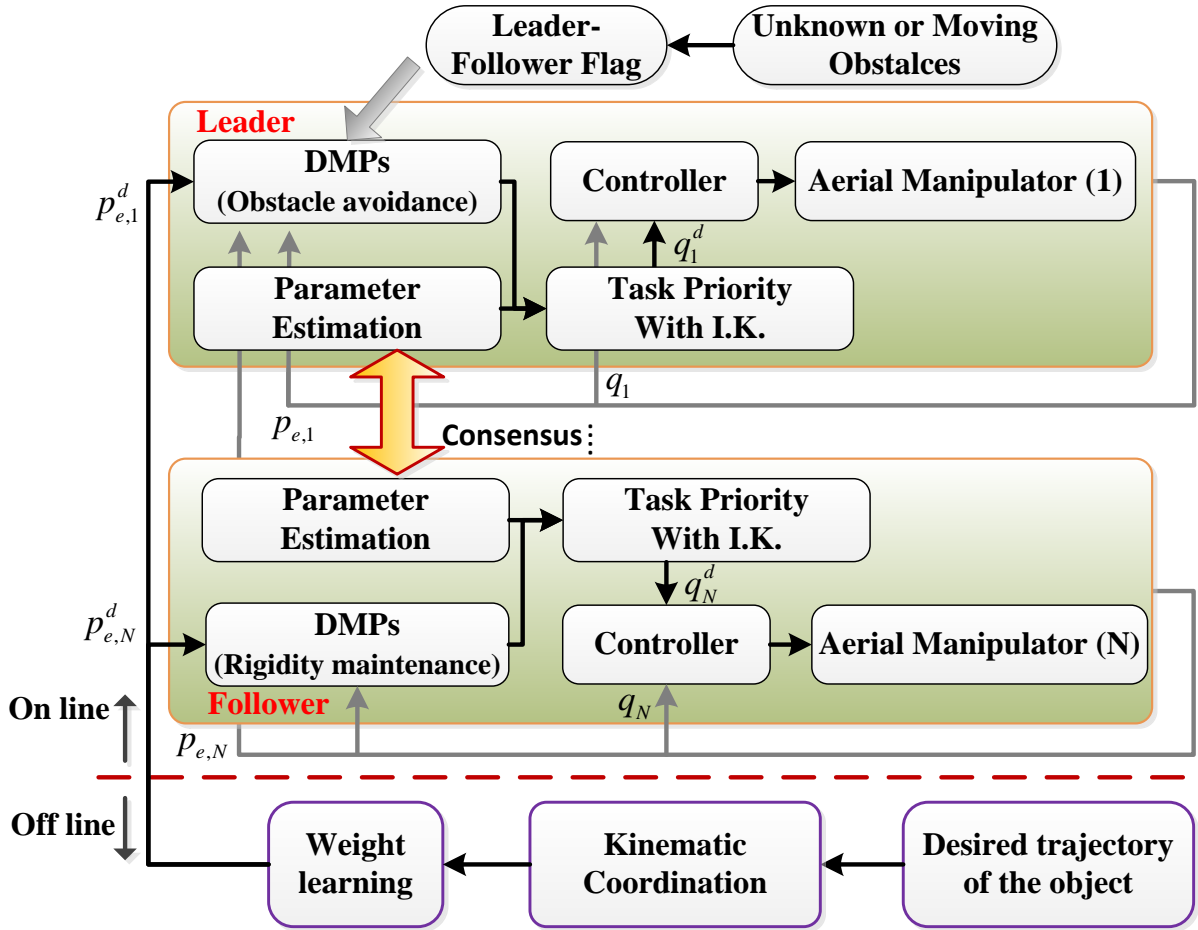


Figure 3.1: Overall structure for the proposed synthesis.

3.1 Payload Mass and Inertia Parameter Estimation

When the end-effector of the robotic arm carries an unknown object, the physical parameters of the combined dynamics in (2.17) are changed because of the unknown mass m_o and inertia J_o . To compensate these unknown effects, this work develops the on-line parameter estimator and designed the controller with the estimated parameters.

3.1.1 System Parametrization

Before achieving aforementioned goal, this work first represents the combined dynamics with estimated parameters as

$$\begin{aligned}\hat{D}_i &= M_i + c_i J_i^T (E_i^\dagger \hat{H}_o E_i^{-T}) J_i \\ \hat{C}_i &= Q_i + c_i J_i^T (E_i^\dagger \hat{\mu}_o E_i^{-T}) J_i + c_i J_i^T (E_i^\dagger \hat{H}_o E_i^{-T}) \dot{J}_i \\ \hat{G}_i &= W_i + c_i J_i^T E_i^\dagger \hat{G}_o,\end{aligned}\tag{3.1}$$

where $\hat{*}$ is the vector or matrix which includes the estimated parameters on the object of the i -th aerial manipulator (i.e., $\hat{m}_{o,i}$ and $\hat{J}_{o,i}$).

In order to derive the parameter estimator, this disseration considers the unknown parameter $\hat{J}_{o,i}$ first. By using proper vision algorithms such as appreared in [67], the target object can be detected based on the 3D CAD model of the object. After matching the unknown object to the CAD model, multiple aerial manipulators can detect and grasp the unknown object by using a proper vision algorithm. This work considers the scenarios after grasping the object, so this is beyond the scope of this paper. Therefore, the following assumption is used.

Assumption 2. *The geometric dimension of the common object is known.*

Based on the assumption 2, for example, the common object is a cylinder with the radius r_c and length h_c , then the unknown inertia $\hat{J}_{o,i}$ can be expressed with respect to the

unknown mass $\hat{m}_{o,i}$ as

$$\hat{\mathbf{J}}_{o,i} = \hat{m}_{o,i} R_o \begin{bmatrix} \mathbf{I}_{ox} & 0 & 0 \\ 0 & \mathbf{I}_{oy} & 0 \\ 0 & 0 & \mathbf{I}_{oz} \end{bmatrix} R_o^T, \quad (3.2)$$

where $\mathbf{I}_{ox} = \mathbf{I}_{oz} = \frac{1}{12}(3r_c^2 + h_c^2)$, $\mathbf{I}_{oy} = \frac{1}{2}r_c^2$ and R_o is the rotation matrix of the object. Therefore, the parameterized equation in (3.1) can be rewritten with respect to $\hat{m}_{o,i}$ as

$$\begin{aligned} \hat{D}_i &= M_i + \hat{m}_{o,i} H_1 \\ \hat{C}_i &= Q_i + \hat{m}_{o,i} H_2 \\ \hat{G}_i &= W_i + \hat{m}_{o,i} H_3, \end{aligned} \quad (3.3)$$

where H_1 , H_2 and H_3 are the matrices with known physical parameters and calculated from (3.1). Finally, by using (3.3), the dynamics in (2.17) can be rewritten as

$$\hat{m}_{o,i}(H_1\ddot{\mathbf{q}}_i + H_2\dot{\mathbf{q}}_i + H_3) = \mathbf{U}_i(t), \quad (3.4)$$

by introducing the forcing term including control input $\boldsymbol{\tau}_i$ in:

$$\mathbf{U}_i(t) = \boldsymbol{\tau}_i - M_i\ddot{\mathbf{q}}_i - Q_i\dot{\mathbf{q}}_i - W_i. \quad (3.5)$$

3.1.2 On-line Parameter Estimator

Now, this work designs the parameter estimator based on the parameterized dynamics in (3.4).

$$\begin{aligned} C^*\dot{\hat{\mathbf{q}}}_i + K^*\hat{\mathbf{q}}_i + \hat{m}_{o,i}(H_1\ddot{\mathbf{q}}_i + H_2\dot{\mathbf{q}}_i + H_3) \\ = \mathbf{U}_i(t) + C^*\dot{\mathbf{q}}_i + K^*\mathbf{q}_i, \end{aligned} \quad (3.6)$$

where $C^* \in \mathbb{R}^{8 \times 8}$ and $K^* \in \mathbb{R}^{8 \times 8}$ are user-defined gain matrices and $\hat{\mathbf{q}}_i$ is the estimated state of the i -th aerial manipulator. For initial parameter update, $\hat{\mathbf{q}}_i(0) \neq \mathbf{q}_i(0)$ is recommended.

If the state estimation error is defined as

$$\mathbf{e}_i = \hat{\mathbf{q}}_i - \mathbf{q}_i, \quad (3.7)$$

then the parameter update rule for $\hat{m}_{o,i}$ can be obtained as

$$\dot{\hat{m}}_{o,i} = \Gamma_1 \mathbf{e}_i^T (H_1 \ddot{\mathbf{q}}_i + H_2 \dot{\mathbf{q}}_i + H_3) + \Gamma_2 \sum_{j=1}^{N_m} (\hat{m}_{o,j} - \hat{m}_{o,i}), \quad (3.8)$$

where Γ_1 and Γ_2 are the learning and consensus rates, respectively. Using (2.17) and (3.6), the error dynamics can be written as

$$C^* \dot{\mathbf{e}}_i + K^* \mathbf{e}_i + \tilde{D}_i \ddot{\mathbf{q}}_i + \tilde{C}_i \dot{\mathbf{q}}_i + \tilde{G}_i = 0, \quad (3.9)$$

where $\tilde{D}_i = \hat{D}_i - D_i$, $\tilde{C}_i = \hat{C}_i - C_i$ and $\tilde{G}_i = \hat{G}_i - G_i$.

If the controller for each aerial manipulator is designed properly (to be discussed in Sec. 3.2), then the state variables will remain bounded.

Lemma 1. *If the state variables \mathbf{q}_i , $\dot{\mathbf{q}}_i$ and $\ddot{\mathbf{q}}_i$ are bounded by the forcing term $\mathbf{U}_i(t)$, then the error dynamics in (3.9) is asymptotically stable.*

Proof. In order to prove the convergence of the error dynamics (3.9), this work first defines the Lyapunov candidate function for all agents as

$$V = \sum_{i=1}^{N_m} V_i, \quad (3.10)$$

where

$$V_i = \frac{1}{2} \mathbf{e}_i^T C^* \mathbf{e}_i + \frac{1}{2\Gamma_1} \tilde{m}_{o,i}^2, \quad (3.11)$$

where $\tilde{m}_{o,i} = \hat{m}_{o,i} - m_{o,i}$ is the estimation error. The time derivative of V_i is given as

$$\begin{aligned}\dot{V}_i &= \mathbf{e}_i^T C^* \dot{\mathbf{e}}_i + \frac{1}{\Gamma_1} \tilde{m}_{o,i} \dot{\tilde{m}}_{o,i} \\ &= -\mathbf{e}_i^T K^* \mathbf{e}_i - \tilde{m}_{o,i} \mathbf{e}_i^T (H_1 \ddot{\mathbf{q}}_i + H_2 \dot{\mathbf{q}}_i + H_3) + \frac{1}{\Gamma_1} \tilde{m}_{o,i} \dot{\tilde{m}}_{o,i},\end{aligned}\tag{3.12}$$

where $\dot{\hat{m}}_{o,i} = \dot{\tilde{m}}_{o,i}$. Now, substituting the update rule (3.8) into (3.12), \dot{V}_i can be rewritten as

$$\dot{V}_i = -\mathbf{e}_i^T K^* \mathbf{e}_i + \frac{\Gamma_2}{\Gamma_1} \tilde{m}_{o,i} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i}),\tag{3.13}$$

where $(\tilde{m}_{o,j} - \tilde{m}_{o,i}) = (\hat{m}_{o,j} - \hat{m}_{o,i})$. Using the fact that

$$\sum_{i=1}^{N_m} \tilde{m}_{o,i} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i}) = -\frac{1}{2} \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i})^2,\tag{3.14}$$

eq. (3.10) can be rewritten as

$$\dot{V} = -\sum_{i=1}^{N_m} \mathbf{e}_i^T K^* \mathbf{e}_i - \frac{\Gamma_2}{2\Gamma_1} \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i})^2 \leq 0.\tag{3.15}$$

This proves the boundedness of \mathbf{e}_i , $\tilde{m}_{o,i}$. We can also say that $\dot{\mathbf{e}}_i$ and $\dot{\tilde{m}}_{o,i}$ are bounded, because $\dot{\mathbf{q}}_i$ and $\ddot{\mathbf{q}}_i$ are bounded in (3.8) and (3.9). Then \ddot{V} is also bounded, which guarantees that the state estimation error (i.e., \mathbf{e}_i) and consensus error (i.e., $|\hat{m}_{o,j} - \hat{m}_{o,i}|$ when $i \neq j$) go to 0 asymptotically by applying Barbalat's lemma. \square

Here, parameter convergence, i.e., $\tilde{m}_{o,i} \rightarrow 0$, follows from [26] when the persistence of excitation is assumed.

3.1.3 Robust Analysis for Measurement Noise

In order to show the robustness of the proposed state estimator, this work considers the dynamics as the following:

$$D_i \ddot{\mathbf{q}}_i + C_i \dot{\mathbf{q}}_i + G_i = \boldsymbol{\tau}_i + \boldsymbol{\Theta}_i,$$

where $\boldsymbol{\Theta}_i \in \mathbb{R}^{(6+n)}$ is a bounded disturbance vector such as measurement noise, which satisfies $\|\boldsymbol{\Theta}_i\| \leq \rho_\Theta$ and $\rho_\Theta > 0$. Then, the error dynamics can be rewritten as

$$C^* \dot{\mathbf{e}}_i + K^* \mathbf{e}_i + \tilde{D}_i \ddot{\mathbf{q}}_i + \tilde{C}_i \dot{\mathbf{q}}_i + \tilde{G}_i - \boldsymbol{\Theta}_i = 0$$

From the analysis of the Lyapunov candidate function in (3.10), the derivative of V in (3.13) can be rewritten as

$$\dot{V}_i = -\mathbf{e}_i^T K^* \mathbf{e}_i + \frac{\Gamma_2}{\Gamma_1} \tilde{m}_{o,i} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i}) + \mathbf{e}_i^T \boldsymbol{\Theta}_i. \quad (3.16)$$

From the same relationship in (3.14), V can be rewritten as

$$\begin{aligned} \dot{V} &= -\sum_{i=1}^{N_m} \mathbf{e}_i^T K^* \mathbf{e}_i - \frac{\Gamma_2}{2\Gamma_1} \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i})^2 + \sum_{i=1}^{N_m} \mathbf{e}_i^T \boldsymbol{\Theta}_i \\ &\leq -\sum_{i=1}^{N_m} \lambda_{\min}(K^*) \|\mathbf{e}_i\|^2 - \frac{\Gamma_2}{2\Gamma_1} \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i})^2 + \sum_{i=1}^{N_m} \|\mathbf{e}_i\| \|\boldsymbol{\Theta}_i\|, \end{aligned} \quad (3.17)$$

where $\lambda_{\min}(K^*)$ is the smallest eigenvalue of the matrix K^* . By applying Young's inequality, i.e., $\mathbf{a}^T \mathbf{b} \leq \|\mathbf{a}\| \|\mathbf{b}\| \leq \frac{1}{2} \|\mathbf{a}\|^2 + \frac{1}{2} \|\mathbf{b}\|^2$ for two vectors \mathbf{a} and \mathbf{b} , \dot{V} in (3.17) can be simplified

as

$$\begin{aligned}
\dot{V} &\leq -\sum_{i=1}^{N_m} \lambda_{\min}(K^*) \|\mathbf{e}_i\|^2 - \frac{\Gamma_2}{2\Gamma_1} \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i})^2 + \frac{1}{2} \sum_{i=1}^{N_m} \|\mathbf{e}_i\|^2 + \frac{1}{2} \sum_{i=1}^{N_m} \|\boldsymbol{\Theta}_i\|^2 \\
&\leq -\sum_{i=1}^{N_m} (\lambda_{\min}(K^*) - \frac{1}{2}) \|\mathbf{e}_i\|^2 - \frac{\Gamma_2}{2\Gamma_1} \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i})^2 + \frac{1}{2} \sum_{i=1}^{N_m} \|\boldsymbol{\Theta}_i\|^2. \tag{3.18}
\end{aligned}$$

At this point, the upper bound of V should be analyzed. From the Lemma 1 of the boundedness of $\tilde{m}_{o,i}$, this work defines ρ_m , which satisfies the following equation: $\sum_{i=1}^{N_m} \tilde{m}_{o,i}^2 < \rho_m$. Note that the upper bound ρ_m goes zero when the persistence of excitation is satisfied. Then, the following equation can be given as

$$\begin{aligned}
V &= \sum_{i=1}^{N_m} \frac{1}{2} \mathbf{e}_i^T C^* \mathbf{e}_i + \sum_{i=1}^{N_m} \frac{1}{2\Gamma_1} \tilde{m}_{o,i}^2 \\
&\leq \frac{1}{2} \sum_{i=1}^{N_m} \lambda_{\max}(C^*) \|\mathbf{e}_i\|^2 + \frac{1}{2\Gamma_1} \sum_{i=1}^{N_m} \rho_m := \bar{V}. \tag{3.19}
\end{aligned}$$

From (3.18), (3.19), and $\sum_{i=1}^{N_m} \|\boldsymbol{\Theta}_i\| \leq \sum_{i=1}^{N_m} \rho_{\Theta} := \rho_{\Sigma}$, the time derivative of V can be yielded as

$$\begin{aligned}
\dot{V} &\leq -2\kappa_1 \bar{V} + \frac{\kappa_1}{\Gamma_1} \sum_{i=1}^{N_m} \rho_m - \frac{\Gamma_2}{2\Gamma_1} \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} (\tilde{m}_{o,j} - \tilde{m}_{o,i})^2 + \frac{1}{2} \rho_{\Sigma}^2 \\
&\leq -2\kappa_1 \bar{V} + \frac{\kappa_1}{\Gamma_1} \sum_{i=1}^{N_m} \rho_m + \frac{1}{2} \rho_{\Sigma}^2 \tag{3.20}
\end{aligned}$$

where $\kappa_1 = \frac{\lambda_{\min}(K^*) - \frac{1}{2}}{\lambda_{\max}(C^*)}$. Therefore, the upper bound, $\bar{\gamma}$ can be set as

$$\frac{\kappa_1 \rho_m N_m}{\Gamma_1} + \frac{1}{2} \rho_{\Sigma}^2 \leq \bar{\gamma}. \tag{3.21}$$

Then, the eq. (3.20) can be simplified as

$$\dot{V} \leq -2\kappa_1 \bar{V} + \bar{\gamma}. \quad (3.22)$$

For any positive constant ρ , if $\kappa_1 \geq \frac{\bar{\gamma}}{2\rho}$, then it can be shown that $\dot{V} \leq 0$ on $V = 2\rho$. Thus, the inequality (3.22) can be solved as

$$0 \leq V(t) \leq \frac{\bar{\gamma}}{2\kappa_1} + (V(0) - \frac{\bar{\gamma}}{2\kappa_1})e^{-2\kappa_1 t}, \forall t \geq 0. \quad (3.23)$$

The eq. (3.23) implies that $V(t)$ is finally bounded by $\frac{\bar{\gamma}}{2\kappa_1}$. Since the size of the compact set, $\frac{\bar{\gamma}}{2\kappa_1}$ depends on the size of $\bar{\gamma}$ and the eigenvalue of C^* and K^* , the upper bound, $\frac{\bar{\gamma}}{2\kappa_1}$ can be made arbitrarily small by adjusting the ratio of C^* and K^* . For detailed analysis, this work includes the simulation results in the end of this section.

3.2 Controller Design

The aerial manipulators are subject to from inevitable external uncertainties such as ground effects or winds from other manipulators. To handle this problem, an adaptive sliding mode controller is designed for each aerial manipulator. The control error is defined as

$$\mathbf{e}_{c,i} = \mathbf{q}_i - \mathbf{q}_i^d, \quad (3.24)$$

where \mathbf{q}_i is the actual state and \mathbf{q}_i^d is the desired state of the i -th aerial manipulator. Then, the sliding surface variable \mathbf{s}_i can be written as

$$\mathbf{s}_i := \dot{\mathbf{q}}_i - \dot{\mathbf{q}}_i^r = \dot{\mathbf{e}}_{c,i} + \Lambda_i \mathbf{e}_{c,i}, \quad (3.25)$$

where $\dot{\mathbf{q}}_i^r = \dot{\mathbf{q}}_i^d - \Lambda_i \mathbf{e}_{c,i}$ and Λ_i is a diagonal gain matrix. Based on (3.25), the control input $\boldsymbol{\tau}_i$ is designed as

$$\boldsymbol{\tau}_i = \hat{D}_i \ddot{\mathbf{q}}_i^r + \hat{C}_i \dot{\mathbf{q}}_i^r + \hat{G}_i - (K_s + \boldsymbol{\delta}_i) \mathbf{s}_i + \boldsymbol{\xi}_i^f + \hat{\Delta}_i, \quad (3.26)$$

where K_s is a diagonal gain matrix and $\hat{\Delta}_i$ is uncertainty estimated by the i -th aerial manipulator. For simplicity, $\boldsymbol{\xi}_i$ is defined as

$$\boldsymbol{\xi}_i := C^* \dot{\mathbf{e}}_i + K^* \mathbf{e}_i \in \mathbb{R}^{8 \times 1}. \quad (3.27)$$

Then, $\boldsymbol{\xi}_i^f$ passes through a first-order filter with time constant α_c as

$$\alpha_c \dot{\boldsymbol{\xi}}_i^f + \boldsymbol{\xi}_i^f = \boldsymbol{\xi}_i, \quad \boldsymbol{\xi}_i^f(0) = \boldsymbol{\xi}_i(0). \quad (3.28)$$

To compensate the skew-symmetry as described in Remark 1, the auxiliary control input $\boldsymbol{\delta}_i$ is defined to handle properties in (2.22) as

$$\boldsymbol{\delta}_i := \frac{c_i}{2} J_i^T E_i^\dagger (\dot{\hat{H}}_o - 2\hat{\mu}) E_i^{-T} J_i, \quad (3.29)$$

where $\dot{\hat{H}}_o$ and $\hat{\mu}$ include estimated value. The update rule for the uncertainty can be given as

$$\dot{\hat{\Delta}}_i = -K_\Delta \mathbf{s}_i, \quad (3.30)$$

where K_Δ a user-defined diagonal matrix.

Lemma 2. *If $\mathbf{e}_i, \dot{\mathbf{e}}_i \in \mathcal{L}_\infty$ and $\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i, \ddot{\ddot{\mathbf{q}}}_i \in \mathcal{L}_\infty$, then $\dot{\boldsymbol{\xi}}_i$ is bounded as follows:*

$$\|\dot{\boldsymbol{\xi}}_i\| \leq \rho_\xi \quad (3.31)$$

where ρ_ξ is a positive constant.

Proof. Consider the following compact sets:

$$\begin{aligned}\mathcal{D}_1 &\triangleq \{(\mathbf{e}_i, \dot{\mathbf{e}}_i) \mid |\mathbf{e}_i|^2 + |\dot{\mathbf{e}}_i|^2 \leq \rho_1\} \\ \mathcal{D}_2 &\triangleq \{(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i, \dddot{\mathbf{q}}_i) \mid |\mathbf{q}_i|^2 + |\dot{\mathbf{q}}_i|^2 + |\ddot{\mathbf{q}}_i|^2 + |\dddot{\mathbf{q}}_i|^2 \leq \rho_2\},\end{aligned}$$

where ρ_1 and ρ_2 are positive constants. From (3.9), $\ddot{\mathbf{e}}_i$ can be rewritten as

$$C^* \ddot{\mathbf{e}} = -K^* \dot{\mathbf{e}}_i + \tilde{D}_i \ddot{\mathbf{q}}_i + \dot{\tilde{D}}_i \dot{\mathbf{q}}_i + \tilde{C}_i \ddot{\mathbf{q}}_i + \dot{\tilde{C}}_i \dot{\mathbf{q}}_i + \dot{\tilde{G}}_i \quad (3.32)$$

where \tilde{D}_i , $\dot{\tilde{D}}_i$, \tilde{C}_i , $\dot{\tilde{C}}_i$ and $\dot{\tilde{G}}_i$ can be computed by (3.3) and (3.8). The right hand side of (3.32) can be seen as a function of $\mathbf{e}_i, \dot{\mathbf{e}}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i$ and $\ddot{\mathbf{q}}_i$, which are all bounded. Therefore, we can say that $\ddot{\mathbf{e}}_i$ is bounded. Since $\ddot{\mathbf{e}}_i$ and $\dot{\mathbf{e}}_i$ are bounded, $\dot{\xi}_i$ is bounded on the compact set $\mathcal{D}_1 \times \mathcal{D}_2$. \square

Now, this work proves the stability of closed-loop dynamics. For simplicity of procedure, the following assumption is used.

Assumption 3. *The desired trajectory is bounded as*

$$|\mathbf{q}_i^d| + |\dot{\mathbf{q}}_i^d| + |\ddot{\mathbf{q}}_i^d| + |\dddot{\mathbf{q}}_i^d| \leq \rho,$$

where ρ is a positive constant.

Theorem 1. *Consider decoupled dynamics of cooperative aerial manipulators in (2.17) with the parameter estimator (3.6). Then, the sliding surface variable for i -th aerial manipulator, \mathbf{s}_i in (3.25), can be made arbitrarily small under the control input $\boldsymbol{\tau}_i$ in (3.26) with the estimated parameters. If, in addition, the state estimation error goes to zero, i.e., $(\dot{\mathbf{e}}_i, \mathbf{e}_i) \rightarrow 0$ or changes very slowly, then \mathbf{s}_i goes to zero asymptotically.*

Proof. Based on (3.26), the closed-loop dynamics is rewritten as

$$D_i \dot{\mathbf{s}}_i + C_i \mathbf{s}_i + (K_s + \boldsymbol{\delta}_i) \mathbf{s}_i = \tilde{D}_i \ddot{\mathbf{q}}_i^r + \tilde{C}_i \dot{\mathbf{q}}_i^r + \tilde{G}_i + \boldsymbol{\xi}_i^f + \tilde{\boldsymbol{\Delta}}_i, \quad (3.33)$$

where $\tilde{\Delta}_i$ is the uncertainty estimation error. The stability of the closed dynamics in (3.33) can be achieved by using the following Lyapunov candidate function:

$$V_i^c = \frac{1}{2} \mathbf{s}_i^T \hat{D}_i \mathbf{s}_i + \frac{1}{2} \tilde{\Delta}_i^T K_{\Delta}^{-1} \tilde{\Delta}_i + \frac{1}{2} \chi_i^T \chi_i, \quad (3.34)$$

where $\chi_i = \xi_i^f - \xi_i$. The time derivative of V_i^c can be obtained as

$$\begin{aligned} \dot{V}_i^c = & \mathbf{s}_i^T D_i \dot{\mathbf{s}}_i + \frac{1}{2} \mathbf{s}_i^T \dot{D}_i \mathbf{s}_i + \mathbf{s}_i^T \tilde{D}_i \dot{\mathbf{s}}_i + \frac{1}{2} \mathbf{s}_i^T \dot{\tilde{D}}_i \mathbf{s}_i \\ & + \tilde{\Delta}_i^T K_{\Delta}^{-1} \dot{\tilde{\Delta}}_i + \chi_i^T \dot{\chi}_i, \end{aligned} \quad (3.35)$$

where $\tilde{D}_i = \hat{D}_i - D_i$ and $\tilde{\Delta}_i = \hat{\Delta}_i - \Delta_i$. From (3.9), we can obtain $\tilde{G}_i = -\xi_i - \tilde{D}_i \ddot{\mathbf{q}}_i - \tilde{C}_i \dot{\mathbf{q}}_i$. Then we can rewrite (3.35) as

$$\begin{aligned} \dot{V}_i^c = & \mathbf{s}_i^T [-C_i \mathbf{s}_i - (K_s + \delta_i) \mathbf{s}_i + \tilde{D}_i \ddot{\mathbf{q}}_i^r - \tilde{D}_i \ddot{\mathbf{q}}_i + \tilde{C}_i \dot{\mathbf{q}}_i^r \\ & - \tilde{C}_i \dot{\mathbf{q}}_i + \xi_i^f - \xi_i + \tilde{\Delta}_i] + \mathbf{s}_i^T \tilde{D}_i \dot{\mathbf{s}}_i + \frac{1}{2} \mathbf{s}_i^T \dot{\tilde{D}}_i \mathbf{s}_i \\ & + \frac{1}{2} \mathbf{s}_i^T \dot{\tilde{D}}_i \mathbf{s}_i + \tilde{\Delta}_i^T K_{\Delta}^{-1} \dot{\tilde{\Delta}}_i + \chi_i^T \dot{\chi}_i. \end{aligned} \quad (3.36)$$

In this case, since $\frac{1}{2} \mathbf{s}_i^T (\dot{\tilde{D}}_i - 2\hat{C}_i) \mathbf{s}_i - \mathbf{s}_i^T \delta_i \mathbf{s}_i = 0$, from (3.29) and (3.30), we can simplify \dot{V}_i^c as

$$\begin{aligned} \dot{V}_i^c = & -\mathbf{s}_i^T \chi_i - \mathbf{s}_i K_s \mathbf{s}_i - \frac{1}{\alpha} \chi_i^T \chi_i - \chi_i^T \dot{\xi}_i, \\ \leq & -\lambda_{\min}(K_s) \|\mathbf{s}_i\|^2 - \frac{1}{\alpha} \|\chi_i\|^2 + \|\mathbf{s}_i\| \|\chi_i\| + \|\chi_i\| \|\dot{\xi}_i\| \end{aligned} \quad (3.37)$$

In the derivation, we use the fact that $\dot{\chi}_i := \xi_i^f - \xi_i = \frac{\mathbf{x}_i}{\alpha} - \dot{\xi}_i$. $\lambda_{\min}(K_s)$ is smallest eigenvalue of the matrix K_s . By using Lemma 2 and Young's inequality, i.e., $\mathbf{a}^T \mathbf{b} \leq \|\mathbf{a}\| \|\mathbf{b}\| \leq \frac{1}{2} \|\mathbf{a}\|^2 +$

$\frac{1}{2}\|\mathbf{b}\|^2$ for two vectors \mathbf{a} and \mathbf{b} , the eq. (3.37) is rewritten as

$$\begin{aligned}
\dot{V}_i^c &\leq -\lambda_{\min}(K_s)\|\mathbf{s}_i\|^2 - \frac{1}{\alpha}\|\boldsymbol{\chi}_i\|^2 + \frac{1}{2}\|\mathbf{s}_i\|^2 + \frac{1}{2}\|\boldsymbol{\chi}_i\|^2 + \frac{1}{2}\|\boldsymbol{\chi}_i\|^2\|\dot{\boldsymbol{\xi}}_i\|^2 + \frac{1}{2} \\
&\leq -(\lambda_{\min}(K_s) - \frac{1}{2})\|\mathbf{s}_i\|^2 - (\frac{1}{\alpha} - \frac{1}{2}\rho_\xi^2)\|\boldsymbol{\chi}_i\|^2 - \frac{1}{2}(1 - \frac{\|\dot{\boldsymbol{\xi}}_i\|^2}{\rho_\xi^2})\rho_\xi^2\|\boldsymbol{\chi}_i\|^2 + \frac{1}{2} \\
&\leq -\kappa_2\|\boldsymbol{\chi}_i\|^2 - \kappa_3\|\mathbf{s}_i\|^2 + \frac{1}{2},
\end{aligned} \tag{3.38}$$

where $\kappa_2 := (\frac{1}{\alpha} - \frac{\rho_\xi^2}{2}) > 0$ and $\kappa_3 := (\lambda_{\min}(K_s) - \frac{1}{2}) > 0$. At this point, let us consider Lyapunov candidate function V_2 :

$$\begin{aligned}
V_i^c &= \frac{1}{2}\mathbf{s}_i^T(\hat{D})_i\mathbf{s}_i + \frac{1}{2}\tilde{\boldsymbol{\Delta}}_i^T K_{\boldsymbol{\Delta}}^{-1}\tilde{\boldsymbol{\Delta}}_i + \frac{1}{2}\boldsymbol{\chi}_i^T\boldsymbol{\chi}_i \\
&\leq \frac{1}{2}\lambda_{\max}(\hat{D}_i)\|\mathbf{s}_i\|^2 + \frac{\|K_{\boldsymbol{\Delta}}^{-1}\|}{2}\|\tilde{\boldsymbol{\Delta}}_i\|^2 + \frac{1}{2}\|\boldsymbol{\chi}_i\|^2 := \bar{V}_i^c.
\end{aligned} \tag{3.39}$$

From (3.38) and (3.39), the time derivative of V_2 is bounded as

$$\begin{aligned}
\dot{V}_i^c &\leq -\kappa_2\|\boldsymbol{\chi}_i\|^2 - \kappa_3\|\mathbf{s}_i\|^2 + \frac{1}{2} \\
&\leq -\beta\|\boldsymbol{\chi}_i\|^2 - \beta\lambda_{\max}(\hat{D}_i)\|\mathbf{s}_i\|^2 + \frac{1}{2} \\
&\leq -2\beta\bar{V}_i^c + \beta\|K_{\boldsymbol{\Delta}}^{-1}\|\|\tilde{\boldsymbol{\Delta}}_i\|^2 + \frac{1}{2}
\end{aligned} \tag{3.40}$$

where

$$0 < \beta < \min \left\{ \kappa_2, \frac{\kappa_3}{\lambda_{\max}(\hat{D})_i} \right\}.$$

From (3.40) and $V_2 \leq \bar{V}_2$, (3.40) yields

$$\begin{aligned}
\dot{V}_i^c &\leq -2\beta\bar{V}_i^c + \beta\bar{\boldsymbol{\Delta}}_i + \frac{1}{2} \\
&\leq -2\beta V_i^c + \gamma_i,
\end{aligned} \tag{3.41}$$

where $\|K_{\Delta}^{-1}\|\|\Delta_i\|^2 \leq \bar{\Delta}_i$ and $\gamma_i = \beta\bar{\Delta}_i + \frac{1}{2}$.

If $\beta > \frac{\gamma}{2\rho}$, then it can be shown that $\dot{V}_i^c \leq 0$ on $V_i^c = 2\rho$. Therefore, $V_i^c \leq 2\rho$ represents an invariant set. Thus, the inequality equation (3.41) implies

$$0 \leq V_i^c(t) \leq \frac{\gamma}{2\beta} + (V_i^c(0) - \frac{\gamma}{2\beta}) \exp^{-2\beta t}, \forall t \geq 0. \quad (3.42)$$

This proves the boundedness of sliding surface \mathbf{s}_i , estimation error of lumped uncertainty $\tilde{\Delta}_i$ and filtered error χ_i . By adjusting γ and β , the upper bound $\gamma/2\beta$ can be made arbitrarily small. Consequently, sliding surface \mathbf{s}_i can be made arbitrarily small.

If estimation error goes zero, i.e., $(\mathbf{e}_i, \dot{\mathbf{e}}_i) \rightarrow 0$, or changes very slowly, we can say that $\xi_i^f \approx \xi_i$, i.e., $\chi_i \approx 0$. In this case, the derivative of Lyapunov candidate function \dot{V}_i^c in (3.35) can be rewritten as

$$\dot{V}_i^c \leq -\lambda_{\min}(k)\|\mathbf{s}_i\|^2. \quad (3.43)$$

As same with (3.15), we can show asymptotic stability of the proposed controller by application of Barbalat's lemma [68]. In this situation, sliding surface \mathbf{s}_i goes zero asymptotically. It also means that $(\mathbf{e}_{c,i}, \dot{\mathbf{e}}_{c,i}) \rightarrow 0$ asymptotically.

In addition, we can say that $\ddot{\mathbf{s}}_i$ is bounded by the derivative of the error dynamics in (3.33) and the assumption 3. It also means that $q^{(3)}$ is also bounded, which is relaxation of Lemma 1. \square

For the total number of aerial manipulators as same with (3.10), new Lyapunov function is defined as

$$V^t = \sum_{i=1}^{N_m} V_i^c. \quad (3.44)$$

From the fact that

$$\dot{V}^t = \sum_{i=1}^{N_m} \dot{V}_i^c < 0, \quad (3.45)$$

Theorem 1 can be applied for the whole system.

3.3 Simulation Results

In order to verify the performance of the proposed estimator and controller, this work presents simulation results of two aerial manipulators. The simulation includes the following results: (i) the effect of the noise and κ_1 and (ii) the comparison results between the proposed controller and conventional passivity-based adaptive control with parameter estimation appeared in [56].

In simulation, the parameters are set as $m_b = 1.1$ kg, mass of each link of the robotic arm as 0.1 kg and the length of each link as 0.2 m. The moment of inertia of the aerial robot is taken from [58] as $I_b = \text{diag}[0.013; 0.013; 0.021]$. The mass of an unknown payload is set as 1.5 kg and the length of the object is 0.8 m.

Gains for the estimator are set as $\Gamma_1 = 0.8$, $\Gamma_2 = 20.0$ in (3.8). The gains for the proposed controller law are set as

$$K_s = \text{diag}[2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 0.5, 0.8] \text{ in (3.26)}$$

$$\Lambda_i = \text{diag}[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0] \text{ in (3.25)}$$

$$K_\Delta = \text{diag}[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5] \text{ in (3.26)}.$$

In the conventional adaptive controller, the gains are set as same with the proposed controller. However, the update rate is set as $\Gamma_1 = 0.1$ because the larger update rate is set,

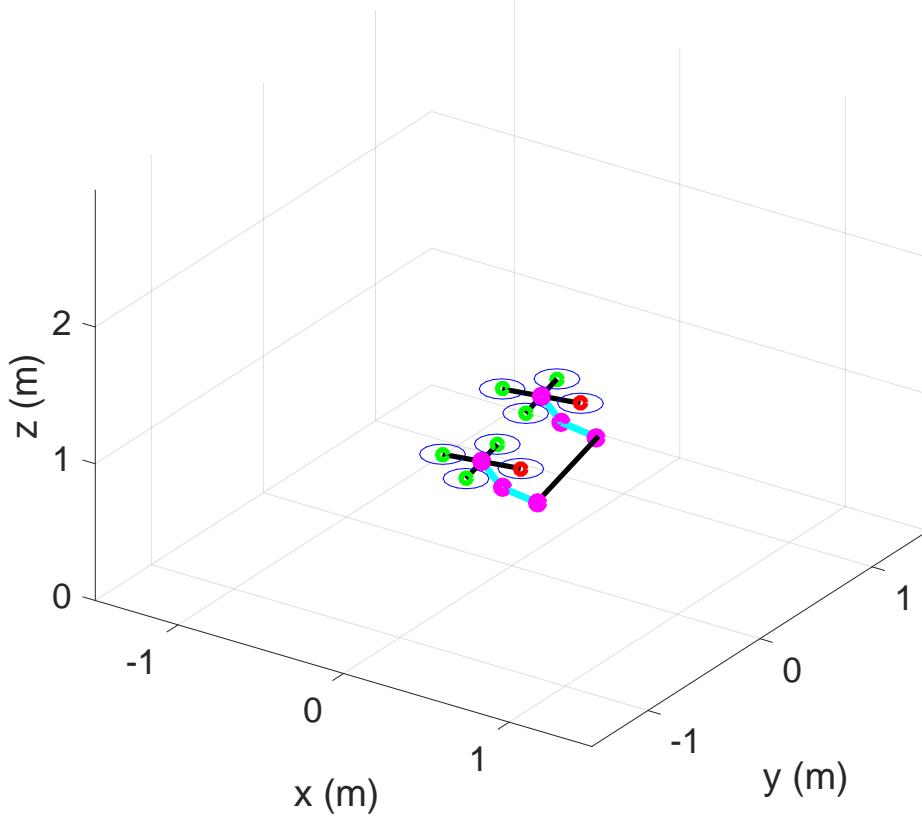


Figure 3.2: Simulation environments

the higher oscillations occur. The desired trajectory for the common object is

$$\mathbf{q}_o^d = \left[\frac{1}{2} \sin\left(\frac{\pi}{10}t\right), \frac{2}{5} \cos\left(\frac{\pi}{10}t\right), \frac{1}{2} \cos\left(\frac{\pi}{10}t\right) \right]^T. \quad (3.46)$$

For simplicity, in this simulation in Fig. 3.2, the conventional inverse kinematics and kinematic coordinations are applied to generate the trajectory of each aerial manipulator as appeared in [7,25].

To analyze the effects of noise and the size of the compact set $\frac{\bar{\gamma}}{2\kappa_1}$ in (3.23), this work performs Monte Carlo simulations consisting of 100 sample runs. Table 3.1 shows the root-mean-square (RMS) performance with respect to the noise level. In this table, noise level means the value of standard deviations of Gaussian noise which is applied to \mathbf{q}_i and 0.1

for $\dot{\mathbf{q}}_i$. From this table, this work can conclude that the size of compact set $\frac{\bar{\gamma}}{2\kappa_1}$ increases as the measurement noise increases. Table 3.2 shows the RMS performance with respect to the κ_1 . From this table, this work also conclude that the size of compact set $\frac{\bar{\gamma}}{2\kappa_1}$ decreases as the value of κ_1 increase.

To analyze the performance of the proposed method compared with the conventional direct adaptive control, Monte Carlo simulations was performed with 100 sample runs. Table 3.3 shows the RMS performance with respect to the noise level. As appeared in this table, the proposed method shows more precise estimation results compared with the direct adaptive method. Fig. 3.3 presents the estimation results of a first trial by the proposed method and conventional passivity-based adaptive controller. In Fig. 3.3, the solid line means the estimation of the first aerial manipulator and dash-dotted line is the results by the second aerial manipulator. From this figure, the proposed estimator shows more clear and faster convergence rate. Thanks to the consensus rule, the difference of parameter estimation is almost zero, while the conventional method shows relatively larger estimation error between two aerial manipulators.

Fig. 3.4 shows the tracking errors of Fig. 3.3 between the actual state and the desired state. In this figure, the tracking performance of the conventional passivity-based adaptive controller shows worse performance than the proposed method. This is mainly because the oscillated estimation results by the conventional method affect the tracking errors, especially in the direction of altitude (i.e., z_o).

Table 3.1: The RMS performance for the estimation with respect to the change of the noise level

Noise level	κ_1	$\tilde{m}_{o,1}$ [kg]	$\tilde{m}_{o,2}$ [kg]	e_x [m]	e_y [m]	e_z [m]
0.10	4.75	0.0027	0.0028	0.0270	0.0546	0.0083
0.25	4.75	0.0102	0.0108	0.0610	0.1260	0.0230
0.50	4.75	0.0335	0.0450	0.1665	0.4299	0.1048

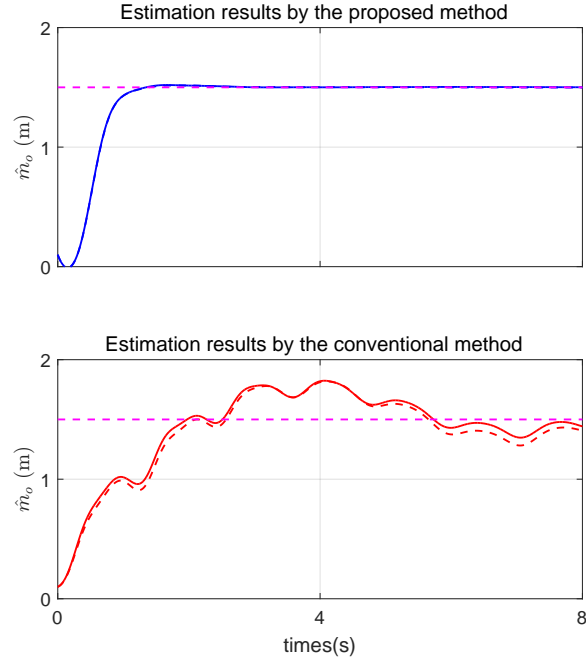


Figure 3.3: Comparison results of the parameter estimation. (Blue line: by the proposed method, Red line: by the conventional method)

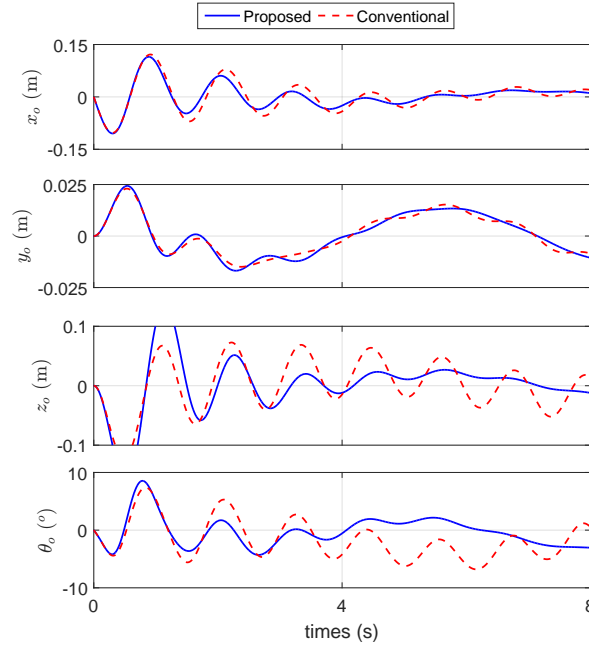


Figure 3.4: Tracking error of the common object.(Blue line: by the proposed method, Red line: by the conventional method)

Table 3.2: The RMS performance for the estimation with respect to the change of κ_1

Noise level	κ_1	$\tilde{m}_{o,1}$ [kg]	$\tilde{m}_{o,2}$ [kg]	e_x [m]	e_y [m]	e_z [m]
0.50	1.75	0.0348	0.0484	0.2723	0.9145	0.1162
0.50	4.75	0.0335	0.0450	0.1665	0.4299	0.1048
0.50	14.75	0.0334	0.0356	0.1153	0.2135	0.0619

Table 3.3: Comparison results of the RMS performance

Noise level	Method	$\tilde{m}_{o,1}$ [kg]	$\tilde{m}_{o,2}$ [kg]	$ \hat{m}_{o,1} - \hat{m}_{o,2} $ [kg]
0.10	Proposed method	0.0027	0.0028	0.0018
0.10	Direct adaptive control	0.1465	0.1207	0.0313
0.25	Proposed method	0.0102	0.0108	0.0050
0.25	Direct adaptive control	0.6319	0.6176	0.0972
0.50	Proposed method	0.0335	0.0450	0.0238
0.50	Direct adaptive control	9.8300	9.9466	0.6615

4

Path Planning

In this section, the desired path generation for cooperative aerial manipulators is addressed. First, the allowable payload is investigated for safe aerial manipulation to consider the maximum thrust of aerial robots. Second, based on the kinematics of the common object and the end-effectors, the desired trajectory of each aerial manipulator is generated by using kinematic coordination and the predefined allowable flight envelope.

4.1 Allowable Payload for Each Aerial Manipulator

Here, this dissertation analyzes the capability of the aerial manipulator with multi-DOF arm with respect to the position of the end-effector. Unlike the preliminary research in [14], this work considers an allowable payload of the aerial manipulator regardless of the DOF of robotic arm. For this, the aerial robot and unknown payload are considered first, in which the torque generated by the robotic arm will be combined later.

In order to operate the hexacopter, the control input $\tau_{b,i}$ in body frame $\Sigma_{b,i}$ from first to sixth elements (i.e., $\tau_{b,i}(1 : 6)$) should be converted into the actuation command from

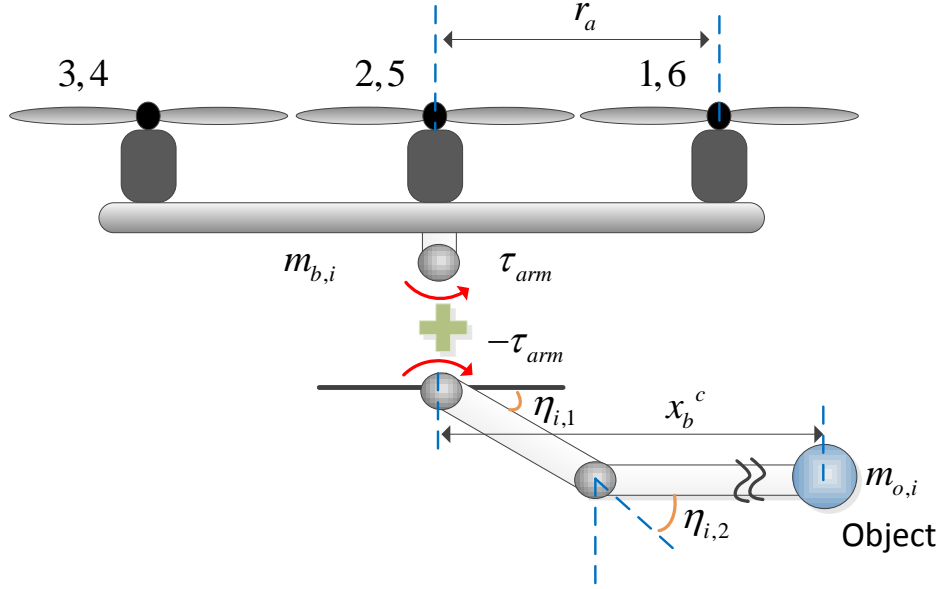


Figure 4.1: Configuration of an aerial manipulator.

(2.9) as

$$\boldsymbol{\tau}_{b,i}(1 : 6) = M_o \mathbf{f}_i. \quad (4.1)$$

Recalling that $\mathbf{f}_i = [f_1, \dots, f_6]$ is the force of the each motor and $f_i = k_f \Omega_i^2$ with the thrust coefficient k_f and the desired speed of the rotor Ω_i . M_o is the motor mapping matrix.

In our configuration, since the robotic arm swings with respect to the y axis in Σ_b , the larger torque can be applied to motor 1 or 6 as shown in Fig. 4.1. Therefore, when the gravitational force due to the maximum allowable payload $m_{o,i}^{\max}$ is acting on the end-effector, the motor command can be calculated as

$$k_f \Omega_{1,\max}^2 = M_o^\dagger(1, :) \boldsymbol{\tau}_{b,i}, \quad (4.2)$$

where $M_o^\dagger(1, :)$ is the first row vector of M_o^\dagger . By using (4.1), the desired speed of motor 1 can be expressed with the mass of a payload ($m_{o,i}^{\max}$) and the length between $\Sigma_{b,i}$ and $\Sigma_{c,i}$

in x direction (x_b^c) as

$$6k_f\Omega_{1,\max}^2 = (m_b + m_{o,i}^{\max})g + \frac{\tau_\phi}{r_a} + \frac{6}{2\sqrt{3}r_a}(\tau_\theta + m_{o,i}^{\max}gx_b^c) + \frac{\tau_\psi}{c_m}, \quad (4.3)$$

where c_m is the ratio between the thrust and drag coefficient, g is the gravitational constant and r_a is the arm length of the hexacopter (i.e., $r_a = 0.2$ m in our configuration). $\Omega_{1,\max}$ is set to 9,200, because the possible range of revolutions per minute (RPM) of the motor is from 1,200 to 9,200. In addition, if the attitude error of the aerial manipulator is bounded by the maximum roll torque, the following equation is satisfied:

$$|K_\phi(\dot{e}_\phi + \Lambda_\phi e_\phi)| \leq K_\phi(\dot{\phi}_{\max} + \Lambda_\phi \phi_{\max}) := \tau_{\phi_{\max}}, \quad (4.4)$$

where K_ϕ and Λ_ϕ are user-defined gains in the ϕ direction. $\tau_{\theta_{\max}}$ and $\tau_{\psi_{\max}}$ can be expressed similarly. Here, $\phi_{\max} = 5$ deg and $\dot{\phi}_{\max} = 14$ deg/s are set. Finally, eq. (4.3) can be rewritten as

$$m_{o,i}^{\max} = \left[r(6k_f\Omega_{1,\max}^2 - m_b g) - \frac{6}{2\sqrt{3}}\tau_{\theta_{\max}} - \frac{r}{c_m}\tau_{\psi_{\max}} - \tau_{\phi_{\max}} \right] / g(r_a + |x_b^c|). \quad (4.5)$$

From the user-defined gains used in the experiments, i.e., $K_\phi = 1.0$ and $\Lambda_\phi = 3.2$, the maximum allowable payload can be calculated by (4.5). Because eq. (4.5) does not consider the disturbances or efficiency of electric motor, 20 % additional margin on $m_{o,i}^{\max}$ are given. As a result, the maximum allowable payload can vary from 0.16 kg to 0.36 kg with respect to the relative position of the object as shown in Fig. 4.2.

However, the above result in Fig. 4.2 does not consider the effect of the robotic arm. Since the torque generated due to the movement of the robotic arm can reduce the allowable payload, this torque should be compensated to obtain the allowable payload for the aerial manipulator. If the torque generated by the robotic arm without a payload is $\tau_{arm,\theta}$ which is applied in the pitch direction of the hexacopter, then the maximum payload with this

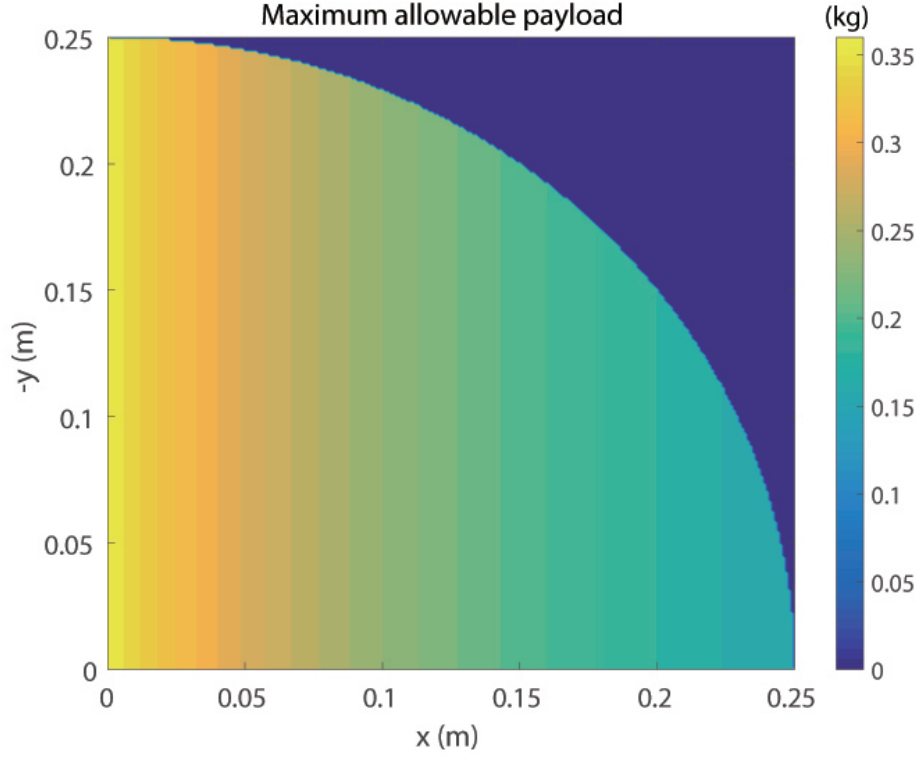


Figure 4.2: Allowable payload with respect to the relative position of an object.

effect can be computed from (4.5) as

$$m_{o,i}^{\max} = m_{o,i}^{\text{act}} + \frac{6|\tau_{\text{arm},\theta}|}{2\sqrt{3}g(r_a + |x_b^c|)} \quad (4.6)$$

where $m_{o,i}^{\text{act}}$ is the actual allowable payload of the hexacopter. For example, if the arm is in the straight forward position (i.e., $x_b^c = 0.25$ m) and $\tau_{\text{arm},\theta} = 0.184$ Nm, the virtual payload due to the robotic arm is about 0.096 kg. So, the final allowable payload range varies from 0.096 kg to 0.36 kg for each aerial manipulator. Based on (4.6), the unilateral constraints can be determined by the total mass that is computed by adding the virtual mass due to the robotic arm to the estimated mass.

4.2 Trajectory Generation with Unilateral Constraints

This section addresses the trajectory generation of each aerial manipulator with unilateral constraints by the allowable flight envelope.

4.2.1 End-effector Trajectory Generation

The trajectory generation for each aerial manipulator consists of two layers : 1) kinematic coordination to generate the trajectory of each end-effector and 2) motion generation with task priority solution. In the first layer, the desired trajectory of each end-effector are computed (i.e., $\mathbf{p}_{e,i}^d$) from the desired trajectory of the object (i.e, $\mathbf{q}_o^d = [\mathbf{p}_o^d, \Phi_o^d]$). From the rigid grasp assumption, it is known that the relative distance from Σ_o and $\Sigma_{e,i}$, which this work denotes as \mathbf{r}_i , is constant. Then, the desired trajectory of the each end-effector can be computed as

$$\begin{aligned}\mathbf{p}_{e,i}^d &= \mathbf{p}_o^d + R_o(\Phi_o^d)\mathbf{r}_i \\ \dot{\mathbf{p}}_{e,i}^d &= \dot{\mathbf{p}}_o^d + S(\omega_o)\mathbf{r}_i,\end{aligned}\tag{4.7}$$

where R_o transforms a vector from frame Σ_o to frame Σ_I .

4.2.2 Inverse Kinematics with Null Space Approach

Before describing the task priority solution, this work first addresses the trajectory generation based on conventional inverse kinematics [54,69]. In this section, a trajectory generation based on inverse kinematics are applied to the aerial manipulator. To restrict the movement of the robotic arm, null space approach is used. Finally, by showing simple experimental result with a aerial manipulator, this work will address the issue of inverse kinematics with null space approach.

A main task of inverse kinematics is to generate the trajectory in joint space \mathbf{q}_i by using the position of the end-effector. To do so, this work considers the kinematic relationship

between the end-effector and the aerial robot. The position of the end-effector (i.e., $\mathbf{p}_{e,i}$) can be obtained by using the Cartesian position of the hexacopter (i.e., $\mathbf{p}_{b,i}$) and the position of the end-effector with respect to $\Sigma_{b,i}$ (i.e., $\mathbf{p}_{e,i}^b$). The kinematic equation between $\mathbf{p}_{e,i}$ and $\mathbf{p}_{b,i}$ can be written as

$$\mathbf{p}_{e,i} = \mathbf{p}_{b,i} + R_{b,i} \mathbf{p}_{e,i}^b \quad (4.8)$$

where $R_{b,i}$ transforms a vector from frame $\Sigma_{b,i}$ to frame Σ_I , $\dot{\mathbf{p}}_{e,i}^b = J_\eta \dot{\boldsymbol{\eta}}_i$, and J_η is the Jacobian matrix. In addition, to control the attitude of the object, the orientation of each end-effector should be considered. Finally, a forward kinematic solution can be established between $\dot{\mathbf{p}}_{e,i}$ and $\dot{\mathbf{q}}_i$ as

$$\begin{aligned} \dot{\mathbf{p}}_{e,i} &= \begin{bmatrix} I_{3 \times 3} & -(R_{b,i} \mathbf{p}_{e,i}^b)^\wedge J_{\Phi_i} & R_{b,i} J_\eta \end{bmatrix} \dot{\mathbf{q}}_i \\ &:= J_{e,i} \dot{\boldsymbol{\nu}}_i + B(\Phi_i) \dot{\Phi}_i \end{aligned} \quad (4.9)$$

where J_{Φ_i} converts $\dot{\Phi}_i$ into the angular velocity in $\Sigma_{b,i}$, and \wedge is the operator that converts a vector into a skew-symmetric matrix. From the definition $\dot{\boldsymbol{\nu}}_i = [\dot{\mathbf{p}}_{b,i}^T, \dot{\boldsymbol{\eta}}_i^T]^T \in \mathbb{R}^5$, finally, the augmented desired position of the end-effector (i.e., $\dot{\mathbf{p}}_l^d$) is calculated as

$$\dot{\mathbf{p}}_l^d := \dot{\mathbf{p}}_{e,i}^d + \kappa(\mathbf{p}_{e,i}^d - \mathbf{p}_{e,i}) - B(\Phi_i) \dot{\Phi}_i \quad (4.10)$$

where $\mathbf{p}_{e,i}^d$ is the desired state of $\mathbf{p}_{e,i}$, $\kappa > 0$ is a diagonal gain matrix. To obtain $\boldsymbol{\nu}_i^d$, inverse kinematics are used for redundant manipulators as

$$\dot{\boldsymbol{\nu}}_i^d = J_{e,i}^\dagger \dot{\mathbf{p}}_l^d + P \dot{\mathbf{q}}_o, \quad (4.11)$$

where $\boldsymbol{\nu}_i^d$ is the desired state of $\boldsymbol{\nu}_i$, $J_{e,i}^\dagger = \Gamma J_{e,i}^T (J_{e,i} \Gamma J_{e,i}^T)^{-1}$, and $\Gamma \in \mathbb{R}^{5 \times 5}$ is a weight matrix and $\dot{\mathbf{q}}_o$ is the homogeneous solution of $J_{e,i} P \dot{\mathbf{q}}_o = 0$, where $P = (I - J_{e,i}^\dagger J_{e,i})$ is a projector in the null space of $J_{e,i}$.

In (4.11) to restrict the motion of the robotic arm, the internal motion, $(I - J_{e,i}^\dagger J_{e,i})\dot{\mathbf{q}}_o$ can be addressed as follows:

$$\dot{\mathbf{q}}_o = \kappa_o \left(\frac{\partial D(\boldsymbol{\nu}_i)}{\partial \boldsymbol{\nu}_i} \right)^T, \quad (4.12)$$

where $\kappa_o > 0$ is a constant and $D(\mathbf{q}_e)$ is a secondary objective function of joint variables defined as

$$D(\boldsymbol{\nu}_i) = -\frac{1}{2 \times 5} \sum_{j=1}^5 \left(\frac{\nu_i(j) - \bar{\nu}_i(j)}{\nu_i(j)^M - \nu_i(j)^m} \right)^2, \quad (4.13)$$

where $\nu_i(j)$ is j th element of the vector $\boldsymbol{\nu}_i$. $\nu_i(j)^M$ and $\nu_i(j)^m$ denote the maximum and minimum joint limit respectively. $\bar{\nu}_i(j) = (\nu_i(j)^M + \nu_i(j)^m)/2$. If we maximize this distance function, redundancy can be exploited to keep the joint variables near $\bar{\nu}_i(j)$. In this paper, $\nu_i(j)^M$ and $\nu_i(j)^m$ are determined automatically based on the allowable flight region by saving the contour information of Fig. 4.3. This figure is calculated based on almost same procedure in (4.5), but the figure is represented with respect to angles of 2-DOF robotic arm. From this figure, if the estimated mass is 0.7 kg, the process for obtaining $\nu_i(j)^M$ and $\nu_i(j)^m$ can be shown in Fig. 4.4. In the contour map, the purple region means the unreachable area due to the arm configuration of the robotic arm.

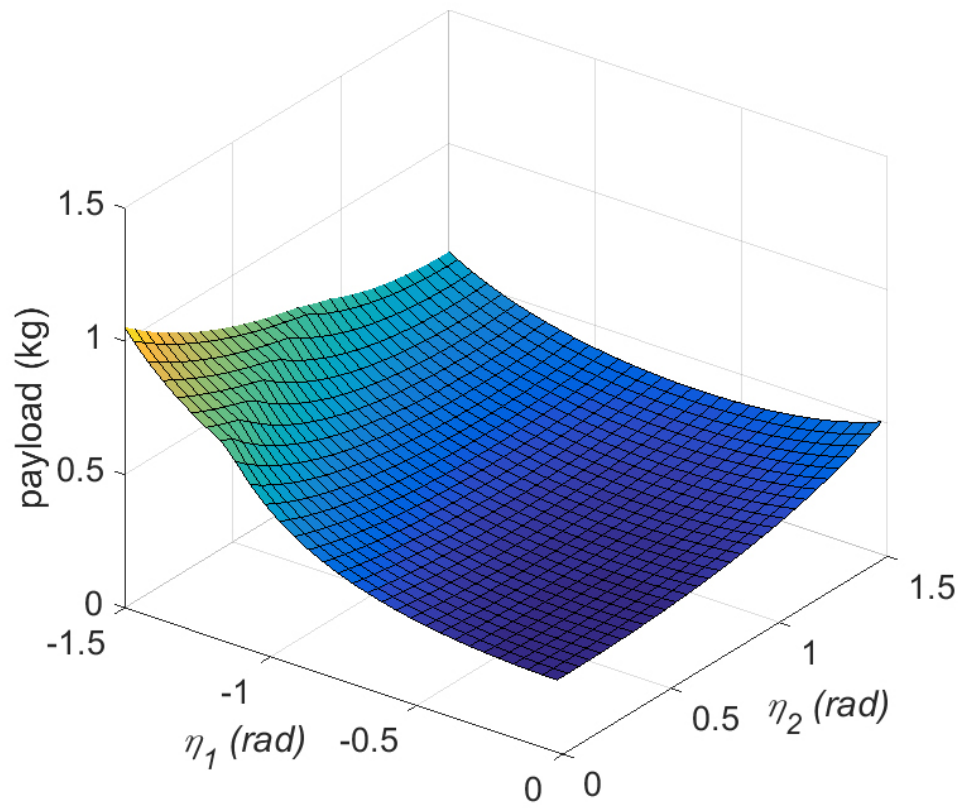


Figure 4.3: The maximum allowable payload according to the arm joint angles. η_i : the joint angle of the link i .

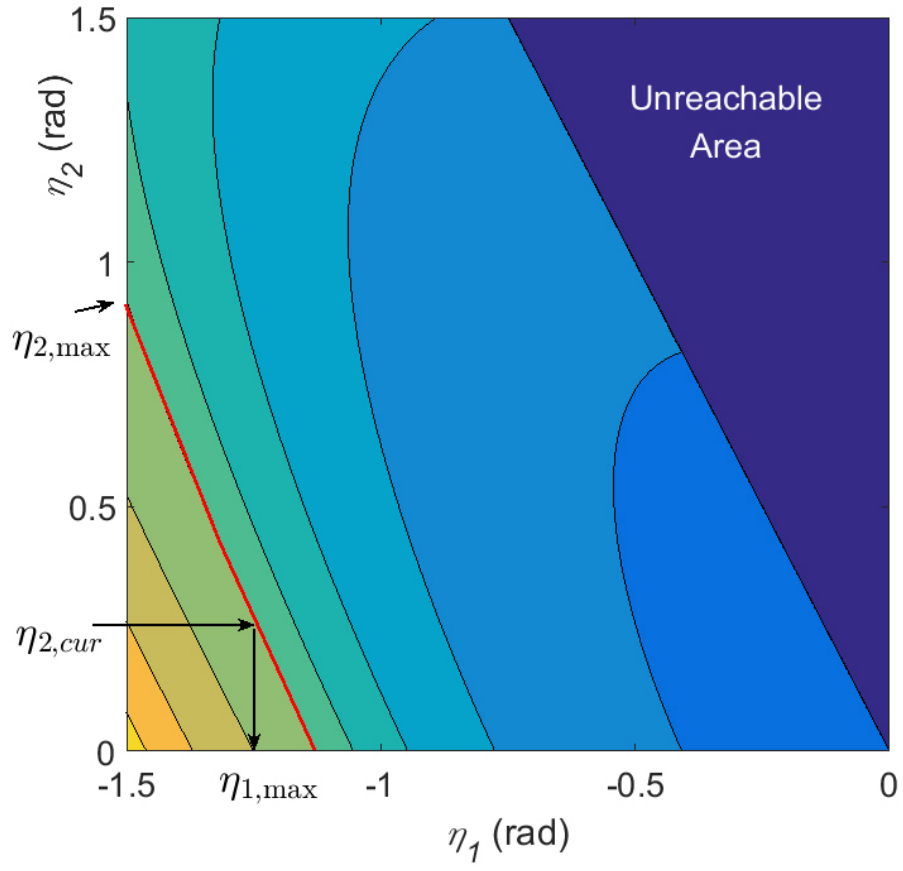


Figure 4.4: Process for computing $\nu_i(j)^M$ and $\nu_i(j)^m$ (Red line : selected flight envelope, $\eta_{i,max}$: maximum joint angle of link i , and $\eta_{i,cur}$: measured joint angle of link i).

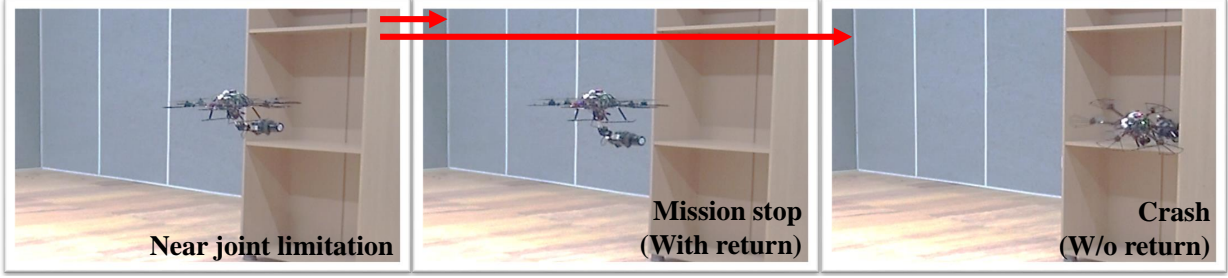


Figure 4.5: Flight with or without return.

In order to validate the aforementioned approach, this work considers the situation when the aerial manipulator delivers a payload to the shelf as shown in Fig. 4.5. To achieve this task, it is desirable to have its arm in almost straight-forward position to avoid the collision with the shelf by adjusting Γ in (4.11). However, when carrying a heavy payload, the aerial robot may collide with the shelf as shown in Fig. 4.5 because of actuation limits. To resolve this problem, this work makes the aerial manipulator come back to the base when the joint angle exceeds the limit as marked with the dotted circle in Fig. 4.6. During the delivery, the aerial manipulator automatically recognizes the potential failure of delivering mission in the phase ② and returns to the base in phase ③. By using our proposed method, the aerial manipulator can return safely to the base.

Despite the satisfactory result, this method is only applicable for the aerial manipulator with 2-DOF robotic arm. It is mainly because the flight envelope with respect to the angle of the robotic arm in Fig. 4.3 is hard to obtain for the robotic arm with higher DOF. To apply the safe transportation method to the robotic arm with higher DOF, unilateral constraints can be a simpler method. In addition, the null space approach does not guarantee that the inequality constraints are always satisfied. Although an aerial manipulator is safe by returning from the mission as shown in Fig. 4.5, the inequality constraints should be satisfied in any circumstances for higher safety for aerial transportation. For these two reasons, this work uses the task priority solution with unilateral constraints, not using the conventional inverse kinematics with the null space approach.

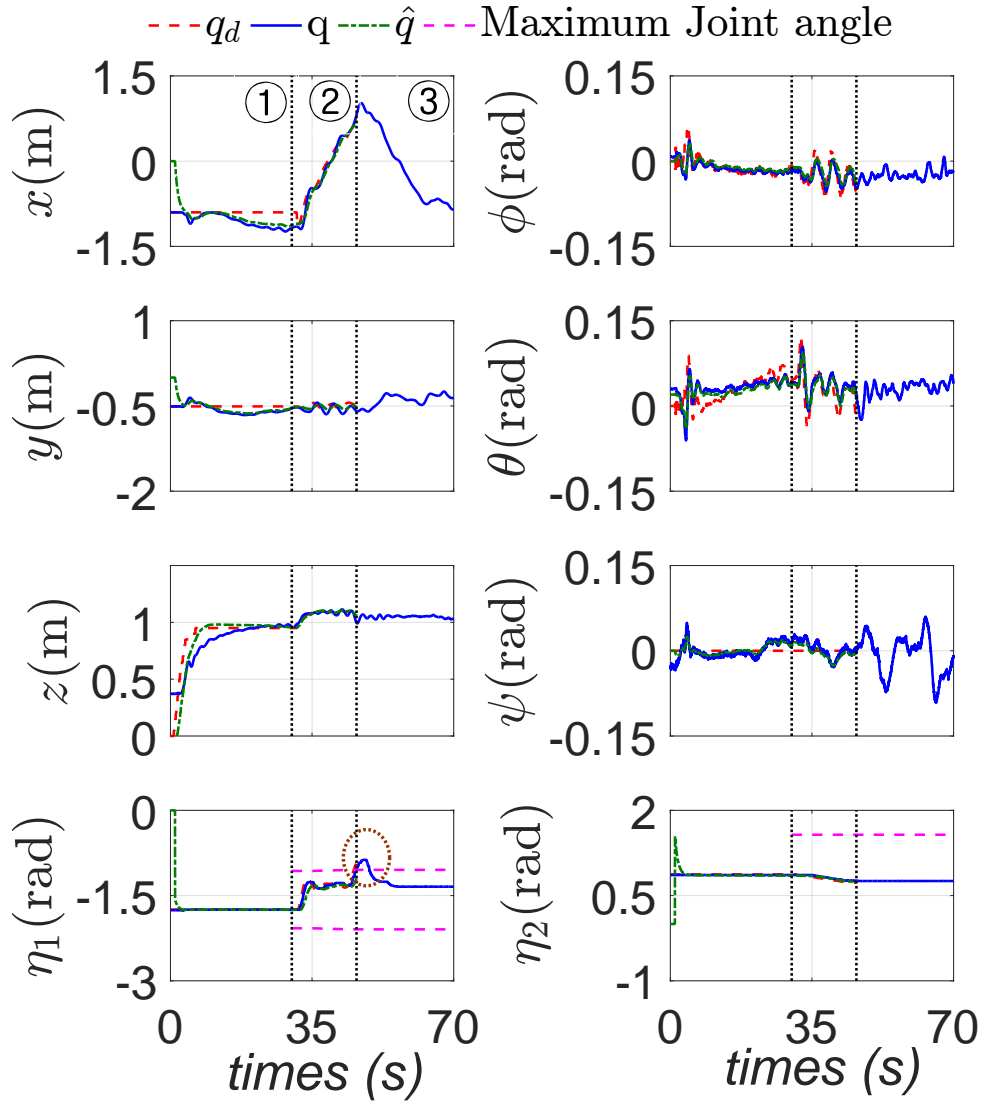


Figure 4.6: Transportation and recovery flight (①: estimation phase, ②: delivery, ③: return to base after mission failure) .

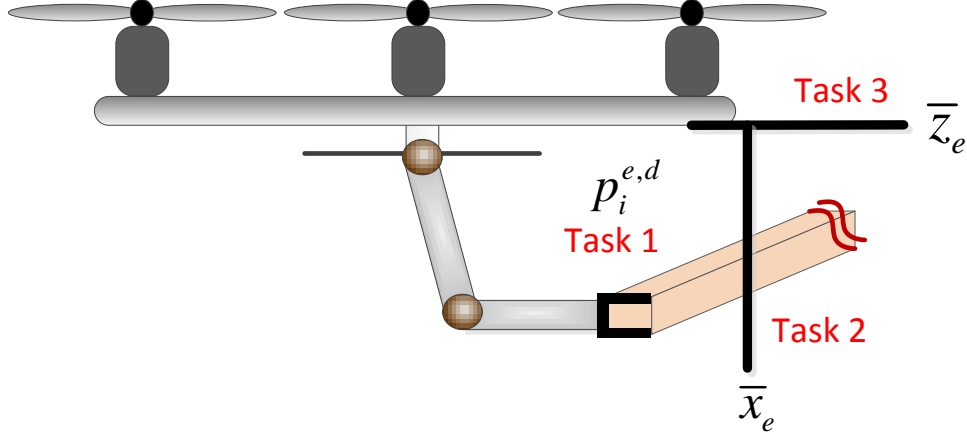


Figure 4.7: Multiple factors for safe aerial transportation. (Task 1: trajectory for transportation, Task 2: unilateral constraints due to the allowable flight envelope, Task 3: unilateral constraints on z axis for the propeller protection.)

4.2.3 Task Prioritization with Unilateral Constraints

The objective of the task priority is to generate the desired trajectory of each aerial manipulator (i.e., \mathbf{q}_i^d) and to track the desired trajectory of the end-effector (i.e., $\mathbf{p}_{e,i}^d$). The end-effector of each aerial manipulator should remain in allowable region as shown in Fig. 4.2, while following the desired trajectory of the object. To achieve this goal, task priority solution are used, which composes of three types of task as shown in Fig. 4.7.

As same with the previous result with inverse kinematics in (4.11), the first priority task is the trajectory generation task for $\boldsymbol{\nu}_{i,1}$. However, in the task priority solution, this work additionally considers that the pitch angle of the object and the pitch of the end-effector should be aligned in the rigid payload, so $\eta_{i,1} + \eta_{i,2} = \theta_o$ are set. Therefore, following the result of [69], this work defines a new task variable for the i -th aerial manipulator as $\boldsymbol{\gamma}_{i,1}^a := [\mathbf{p}_{e,i}; \cos(\eta_{i,1} + \eta_{i,2})]^T$ in (2.27). Finally, a forward kinematic solution can be

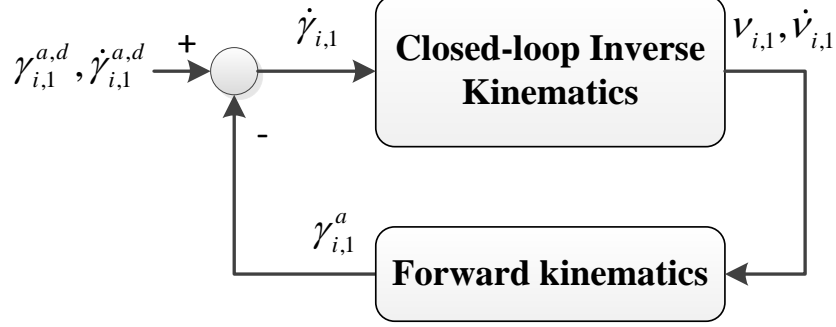


Figure 4.8: Structure for the augmented desired position of the end-effector

established between $\dot{\gamma}_{i,1}^a$ and $\dot{\mathbf{q}}_i$ as

$$\begin{aligned} \dot{\gamma}_{i,1}^a &= \begin{bmatrix} I_{3 \times 3} & -(R_{b,i} \mathbf{p}_{e,i}^b)^\wedge J_{\Phi_i} & R_{b,i} J_\eta \\ 0_{1 \times 3} & 0_{1 \times 3} & J_s \end{bmatrix} \dot{\mathbf{q}}_i \\ &:= T_{i,1} \dot{\nu}_{i,1} + \begin{bmatrix} B(\Phi_i) \\ 0_{1 \times 3} \end{bmatrix} \dot{\Phi}_i \end{aligned} \quad (4.14)$$

where $J_s = [-\sin(\eta_{i,1} + \eta_{i,2}), -\sin(\eta_{i,1} + \eta_{i,2})]$, J_{Φ_i} converts $\dot{\Phi}_i$ into the angular velocity in $\Sigma_{b,i}$, and \wedge is the operator that converts a vector into a skew-symmetric matrix. Finally, the augmented desired position of the end-effector (i.e., $\gamma_{i,1}$) is calculated as

$$\dot{\gamma}_{i,1} = \dot{\gamma}_{i,1}^{a,d} + \kappa(\gamma_{i,1}^{a,d} - \gamma_{i,1}^a) - \begin{bmatrix} B(\Phi_i) \\ 0_{1 \times 3} \end{bmatrix} \dot{\Phi}_i \quad (4.15)$$

where $\gamma_{i,1}^{a,d}$ is the desired state of $\gamma_{i,1}^a$, $\kappa > 0$ is a diagonal gain matrix (See Fig. 4.8). To obtain $\nu_{i,1}$, inverse kinematics are used for redundant manipulators [69] as

$$\dot{\nu}_{i,1} = T_{i,1}^\dagger \dot{\gamma}_{i,1}. \quad (4.16)$$

However, the solution in (4.16) cannot guarantee the safe aerial transportation. To

satisfy the unilateral constraint, task priority solution are used as described in [70]. Based on the allowable flight envelope, the end-effector should not violate the unilateral constraints $\bar{x}_{e,i}$ which can vary depending on the estimated mass. The second task can be obtained based on (2.27) and (4.16) as

$$\dot{\boldsymbol{\nu}}_{i,2} = \dot{\boldsymbol{\nu}}_{i,1} + h_1(T_{i,2}P_{i,1}^A)^\dagger(\dot{\boldsymbol{\gamma}}_{i,2} - T_{i,2}\dot{\boldsymbol{\nu}}_{i,1}) \quad (4.17)$$

Here, $T_{i,2}$ is the transformation matrix of the second task and $P_{i,1}^A = I_{5 \times 5} - T_{i,1}^\dagger T_{i,1}$. In (4.17), the discontinuity near boundary of $\bar{x}_{e,i}$ can be occur, which may cause degradation of tracking performance (See more detail in [70,71]). To resolve this problem, the smooth activation function in h_1 are used. The smooth activation function about boundary condition can be defined as

$$h_1 \begin{cases} 0 & x_e < \bar{x}_e - b_p \\ g(\frac{x_e - \bar{x}_e + b_p}{b_p}) & \bar{x}_e - b_p \leq x_e < \bar{x}_e \\ 1 & \bar{x}_e \leq x_e \end{cases}, \quad (4.18)$$

where b_p is a deactivation buffer and $g(a) = 6a^5 - 15a^4 + 10a^3$ is a quintic polynomial function which satisfies $g(0) = 0$ and $g(1) = 1$.

If the end-effector exceeds the z axis limit (i.e., \bar{z}_e), the collision between the end-effector and propellers can be occurred. For this reason, the third task considers the unilateral constraints on the z axis of the end-effector. Following the same process as (4.17), $\dot{\boldsymbol{\nu}}_{i,3}$ can be computed as

$$\dot{\boldsymbol{\nu}}_{i,3} = \dot{\boldsymbol{\nu}}_{i,2} + h_2(T_{i,3}P_{i,2}^A)^\dagger(\dot{\boldsymbol{\gamma}}_{i,3} - T_{i,3}\dot{\boldsymbol{\nu}}_{i,2}), \quad (4.19)$$

where $T_{i,3}$ is the Jacobian matrix of the third task and $P_{i,2}^A = P_{i,1}^A - (T_{i,2}P_{i,1}^A)^\dagger T_{i,2}P_{i,1}^A$. The activation function h_2 can be computed same as (4.18). Finally, using $\dot{\boldsymbol{\nu}}_{i,3}$, \mathbf{q}_i^d is reconstructed

as

$$\mathbf{q}_i^d = [\boldsymbol{\nu}_{i,3}(1 : 3), \phi_d, \theta_d, \psi_d, \boldsymbol{\nu}_{i,3}(4 : 5)]^T. \quad (4.20)$$

In the rigid grasp, yaw angles of object and aerial manipulators are aligned also, i.e., $\psi_i = \psi_o$, so ψ_d is set to be same with the desired yaw angle of the object.

Note that if the second or third task is near singularity, the damped solution can deform the original task such as $\dot{\boldsymbol{\nu}}_{i,1}$. To resolve issue, the reverse priority approach can be applied [64]. However, in our configuration, since h_1 and h_1 are set to be zero near the corresponding singularity of the robotic arm, the task deformation does not affect the transportation performance. Therefore, although this work uses standard task priority solution in [63], there will be no deformation of the high priority task.

5

Obstacle Avoidance in Unknown Environments

In this section, obstacle detection and avoidance algorithm for cooperative aerial manipulators are described. For obstacle detection, this work uses RGB-D camera. To avoid an unknown obstacle, DMPs are employed for cooperative robots.

5.1 Obstacle Detection

In this work, an RGB-D camera is used to detect an unknown obstacle. This work uses Intel RealSense RGB-D camera. The detailed information for RGB-D cameras is described in table 5.1. From the research in [72], it is known that Intel RealSense camera is very light but noisier. This camera has more missing values than other RGB-D cameras. Nevertheless, since aerial robots do not have enough capability for carrying a heavy camera, RealSense camera is a appropriate solution for obstacle avoidance of aerial robots. For noise in depth measurement, to overcome this problem, this work uses voxel-grid filter and statistical filter from Point Cloud Library (PCL) as described in [52].

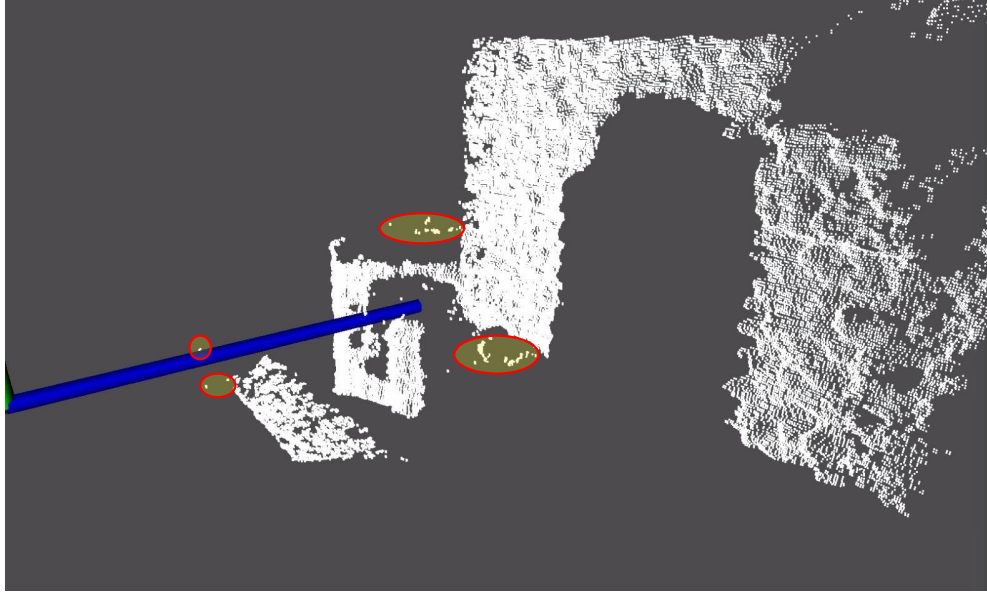


(a) Color image

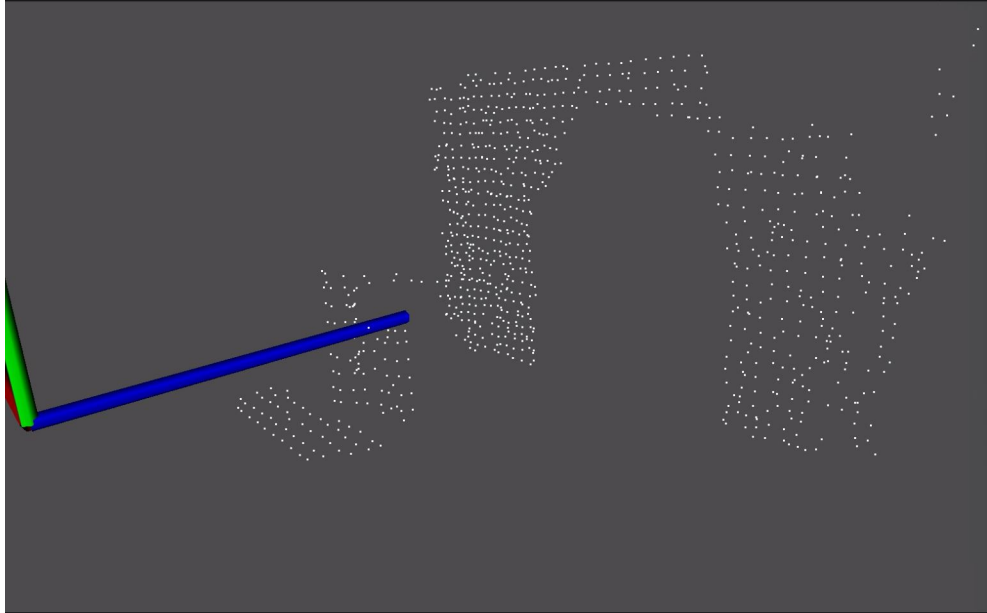


(b) Depth image aligned to the color image

Figure 5.1: Images from RGB-D camera.



(a) 3D reconstruction without the outlier removal filters



(b) 3D reconstruction with the outlier removal filters

Figure 5.2: Outlier rejection by using voxel-grid and statistical filters. (Red circle means the outliers.)

Table 5.1: List of RGB-D Camera

	Realsense	Xtion	Kinect V2
Weight (kg)	0.03	0.23	2.0
power	2.5W USB	2.5W USB	115W
depth resolution	640×480 (60 fps)	640×480 (30 fps)	512×424 (30 fps)
color resolution	1920×1080 (30 fps)	640×480 (30 fps)	1920×1080 (30 fps)
price (\$)	99	Not for sale	99

A voxel grid represents a set of tiny 3D boxes in space. By using a voxel grid, the point clouds can be downsampled with a grid size of 5 mm to speed up the computations. To remove the outliers of the depth measurement completely, this work uses the statistical filter which computes the mean distance of specific point with all neighbors. By assuming that the distribution is Gaussian with a mean and a standard deviation, all points whose mean distances are outside an interval defined by the global distances mean and standard deviation can be treated as outliers and trimmed from the true data.

The results are shown in Fig. 5.1 and Fig. 5.2. Fig. 5.1 shows images obtained from RealSense RGB-D camera. In order to reconstruct 3D point cloud from depth measurement, this work uses depth data aligned to the color image. So, in order to reconstruct 3D point cloud, intrinsic parameters of the camera is used. Fig. 5.2 presents the comparison results with or without the outlier removal filter. In this figure, the blue rod means the coordinate with respect to the camera. From these figures, this work can conclude that outliers of depth measurement are completely removed. Finally, the aerial manipulator detects an unknown obstacle in reconstructed 3D point cloud based on the relative distance from the camera and an unknown obstacle.

5.2 Movement Primitives for Cooperative Aerial Manipulators

In this section, DMPs for cooperative aerial manipulators are developed to avoid an unknown obstacle during transportation. Before doing so, DMPs will learn the predefined trajectory of each end-effector generated by the kinematic coordination.

If two drones avoid each other in z direction, for example, the flight performance can be worse because of downwash from other aerial robots. For this reasons, in this dissertation, two dimensional spaces in horizontal plane are considered simply as $\mathbf{x}_i = [x_{e,i}, y_{e,i}]^T$ for the i -th aerial manipulator. Here $x_{e,i}$ and $y_{e,i}$ are the position of the end-effector in x and y direction, respectively. Although this work uses two dimensional space for obstacle avoidance, the proposed algorithms can be easily extended for avoiding a vertically long obstacle or more generalized obstacle such as other drone or moving birds.

To maintain the distance between two robots, this work uses the leader-follower structure for cooperative DMPs, which is commonly used in multiple mobile robots [8,49]. In aerial robots, the oscillated actual states can cause performance degradation of formation rigidity when all robots want to maintain their desired distance. To handle this problem, we employ the leader-follower structure for cooperative aerial manipulators to avoid oscillated desired trajectory.

Finally, the equation for cooperative DMPs can be written as

$$\begin{aligned}\dot{\mathbf{v}}_i &= K_i(\mathbf{g}_i - \mathbf{x}_i) - B_i\mathbf{v}_i + K_i\mathbf{f}_i(\alpha_i) + (1 - \zeta_{f,i})\mathbf{d}_i(\mathbf{x}) + \zeta_{f,i}\varphi_i(\mathbf{x}_c, \mathbf{o}) \\ \dot{\mathbf{x}}_i &= \mathbf{v}_i \\ \dot{\alpha}_i &= -\gamma\alpha_i,\end{aligned}\tag{5.1}$$

where \mathbf{o} is $x - y$ position of the unknown obstacle in Σ_I detected by RGB-D camera and 3D point cloud, \mathbf{g}_i is goal position of each end-effector and $\zeta_{f,i}$ is the relation parameter between leader and follower. This means that the leader avoids the unknown obstacle and

the follower maintains the rigidity constraint with the leader. For example, if an obstacle is nearby the agent No. 1, then the relation parameters are set as $\zeta_{f,1} = 1$ and $\zeta_{f,2} = 0$. The opposite applies as well. If cooperative aerial manipulators finish avoiding an unknown obstacle, the parameters are set as $\zeta_{f,i} = 1$ for all aerial robots. In this case, all robots follow their own desired trajectory only, because $\varphi_i(\mathbf{x}_c, \mathbf{o}) \approx 0$ when away from the unknown obstacle. In eq. (5.1), the forcing vector \mathbf{f}_i for the i -th aerial manipulator can be defined as

$$\mathbf{f}_i(\alpha_i) = \frac{\sum_{k=1}^{N_w} \alpha_i \Psi_k(\alpha_i) \mathbf{w}_k}{\sum_{k=1}^{N_w} \Psi_k(\alpha_i)}, \quad (5.2)$$

with $\mathbf{w}_k \in \mathbb{R}^{2 \times 1}$ is the weight of the k -th basis function.

From the initial phase to final phase, the weight matrix is obtained as $W_p = [\mathbf{w}_1, \dots, \mathbf{w}_{N_w}]^T \in \mathbb{R}^{N_w \times 2}$ using a least-square sense applying [49]. By discretizing the desired trajectory of each end-effector, the phase variable for i -th aerial manipulator (i.e., α_i) can be calculated.

In the rigid grasp assumption, if multiple robots did not maintain the distance between robots, higher internal forces could be applied to the objects [50]. In addition, if the follower robot follows its own desired trajectory with rigidity constraint, the leader could not avoid unknown obstacle because of repulsive force generated by the follower. This situation will be shown in the experimental results in section 5. To resolve this problem, this work will use simple gradient solution for computational simplicity. To do so, we define the following cost function

$$V_{lf}^a(\mathbf{x}_l, \mathbf{x}_i) = \frac{1}{2}(\|\mathbf{x}_l - \mathbf{x}_i\| - d_{lf})^2 \quad (5.3)$$

where \mathbf{x}_l and \mathbf{x}_i are the horizontal state of the leader and the i -th follower, respectively. d_{lf} is the desired distance between the leader and the follower aerial robots. Then, we can

compute \mathbf{d}_i for the i -th follower robots by the negative gradient solution as

$$\begin{aligned}\mathbf{d}_i(\mathbf{x}) &= -\delta_{lf}^a \frac{\partial V_{lf}^a}{\partial \mathbf{x}_i} \\ &= -\delta_{lf}^a \frac{\mathbf{x}_l - \mathbf{x}_i}{\|\mathbf{x}_l - \mathbf{x}_i\|} (\|\mathbf{x}_l - \mathbf{x}_i\| - d_{lf}),\end{aligned}\tag{5.4}$$

where δ_{lf}^a is the rigidity gain matrix.

In DMPs for moving obstacle avoidance as presented in [48,49], the end-effectors are considered as a point-mass model. However, since aerial manipulator has propellers that can be easily broken in crash, it is important to take the size of aerial robots and object into account. To do so, the size of the boundary r_s is defined first as

$$r_s = \|\mathbf{r}_i\| + k_a d_a,\tag{5.5}$$

where $\|\mathbf{r}_i\|$ is the radius of common object, d_a is the size of aerial manipulator and k_a is the safety margin parameter. In general, since it is difficult to estimate the size of unknown flying obstacles, k_a is set to be larger than 1. Using this value r_s , we define the potential function between center point of multiple aerial manipulators and the unknown object as

$$U(\mathbf{x}_c, o) = \begin{cases} \frac{r_s}{\|\mathbf{x}_c - \mathbf{o}\|}, & \text{if } r_s \leq \|\mathbf{x}_c - \mathbf{o}\| \leq 2r_s \\ 0, & \text{otherwise} \end{cases}.\tag{5.6}$$

When $r_s = 1$, the potential function can be shown as Fig. 5.3. By the negative gradient for $U(\mathbf{x}_c, o)$, we obtain $\varphi(\mathbf{x}_c, \mathbf{o})$ as

$$\begin{aligned}\varphi(\mathbf{x}_c, \mathbf{o}) &= -\frac{\partial U(\mathbf{x}_c, \mathbf{o})}{\partial \mathbf{x}_c} \\ &= \begin{cases} \frac{r_s}{\|\mathbf{x}_c - \mathbf{o}\|^3}(\mathbf{x}_c - \mathbf{o}), & \text{if } r_s \leq \|\mathbf{x}_c - \mathbf{o}\| \leq 2r_s \\ 0, & \text{otherwise} \end{cases},\end{aligned}\tag{5.7}$$

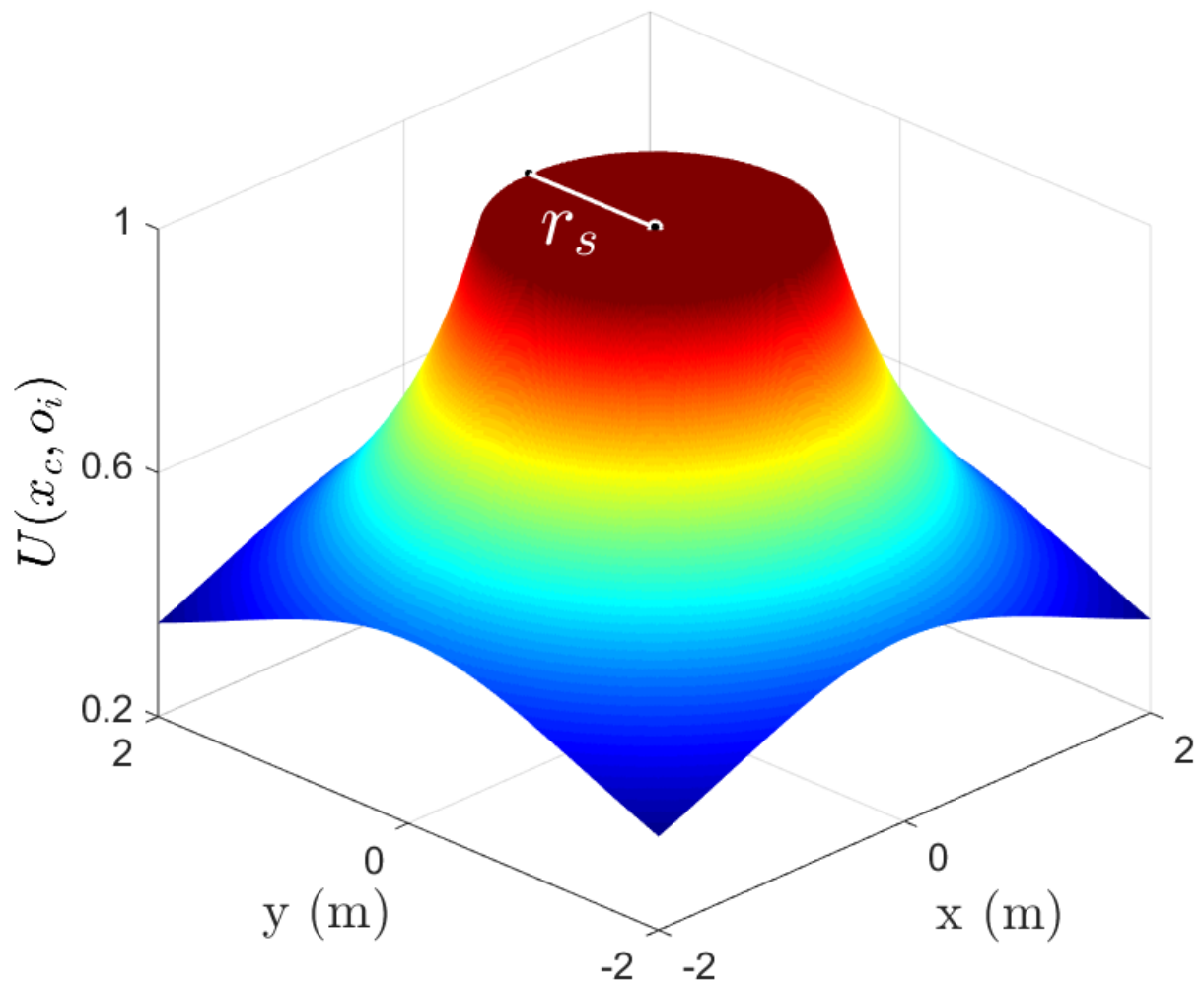


Figure 5.3: Potential Function considering size of robots.

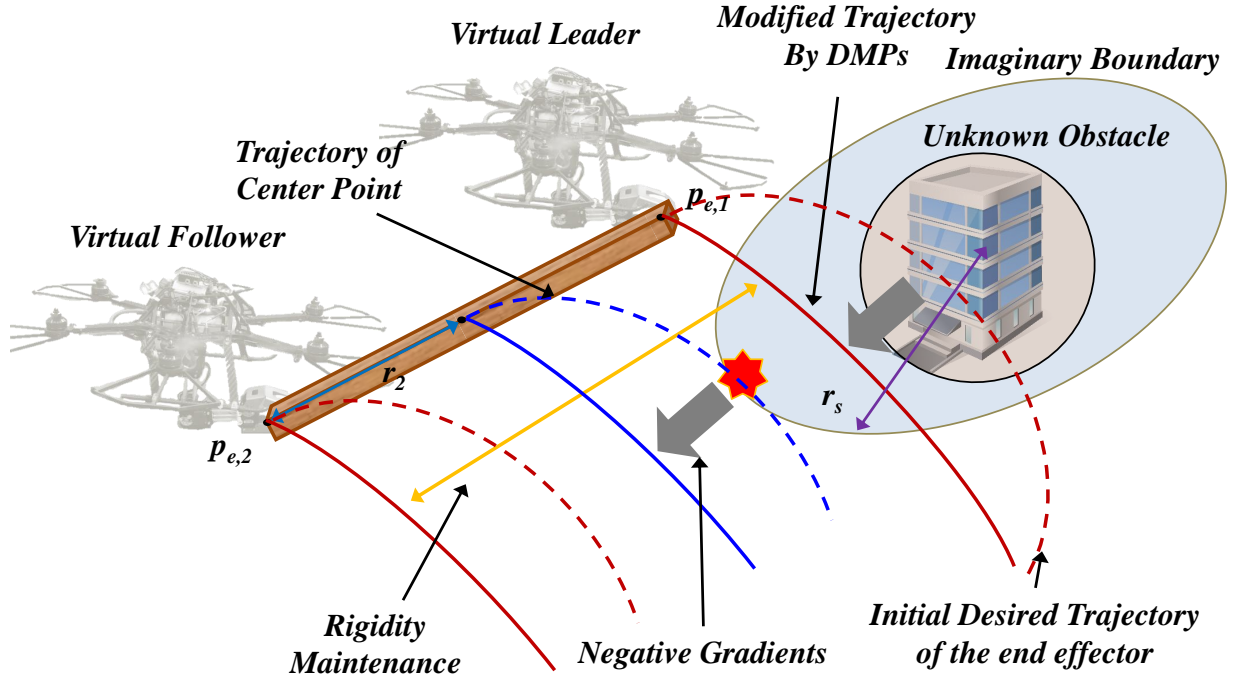


Figure 5.4: Virtual leader and follower structure to avoid unknown obstacles.

Finally, since the attracting systems (5.1) are modified by \mathbf{d}_i and $\varphi(\mathbf{x}_c, \mathbf{o})$, we should investigate the equilibria of the attracting system. In general, the unknown obstacle moves away from the leader as time goes infinity, i.e., $\varphi(\mathbf{x}_c, \mathbf{o}) \approx 0$. The leader will converge to the target position as same with [48]. Then, the follower also converges to the target position since the follower modifies the trajectory only using \mathbf{d}_i , not using $\varphi(\mathbf{x}_c, \mathbf{o})$.

Using \mathbf{d}_i and $\varphi(\mathbf{x}_c, \mathbf{o})$, cooperative aerial manipulators avoid the unknown obstacle considering the size of object and robots using the proposed DMPs as shown in Fig. 5.4. The leader robot computes repulsive force between the imaginary boundary of the unknown obstacle (r_s) and center point \mathbf{x}_c . Then, the follower maintains the distance of the leader by receiving the modified trajectory of the leader. With the proposed algorithm, finally, the aerial manipulators avoid the unknown obstacle.

Simulation results for two and three agents are shown in Fig. 5.5-5.6. The blue, red, and green dashed line mean the previously planned trajectory and the blue, red, and green solid line mean the modified trajectory by DMPs to avoid unknown obstacles. As shown

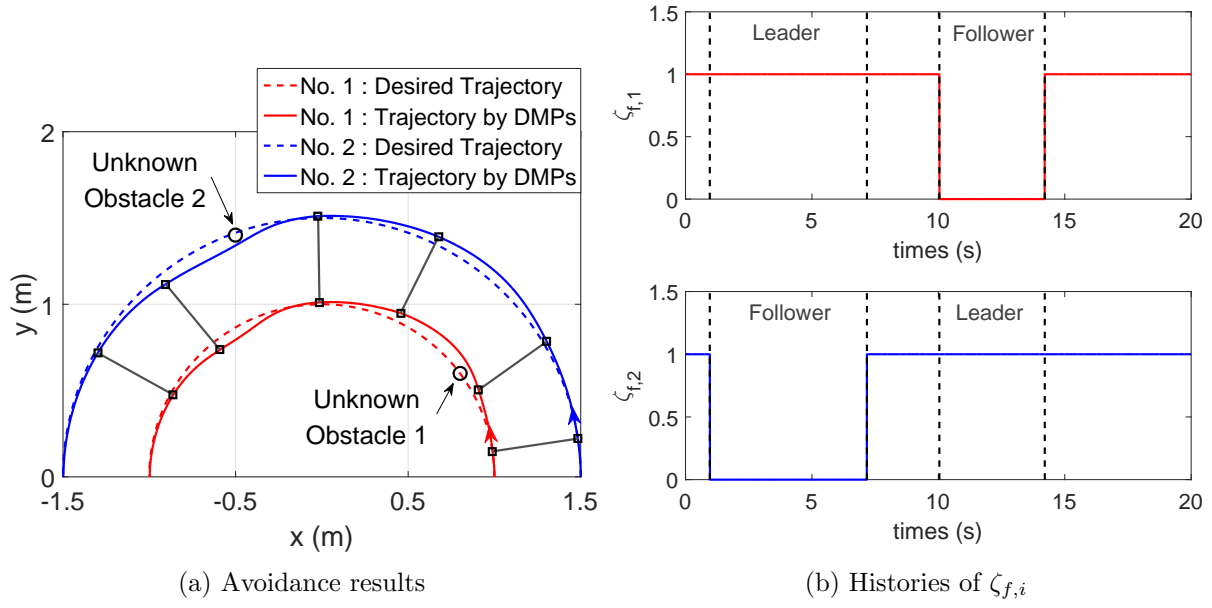
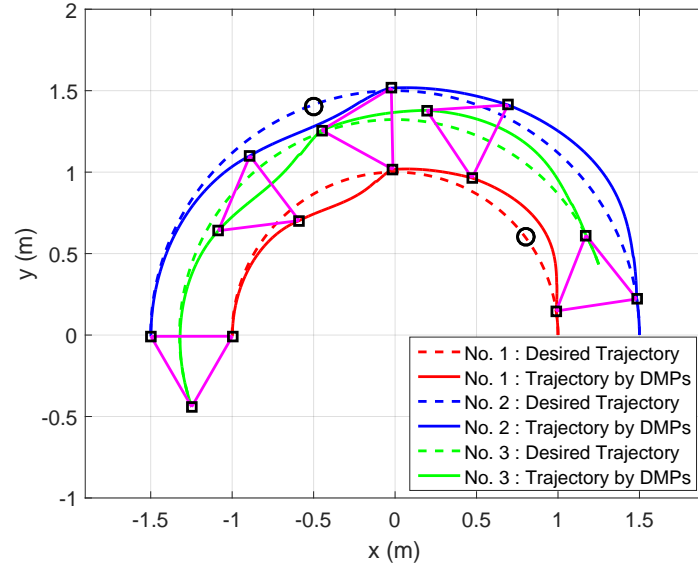
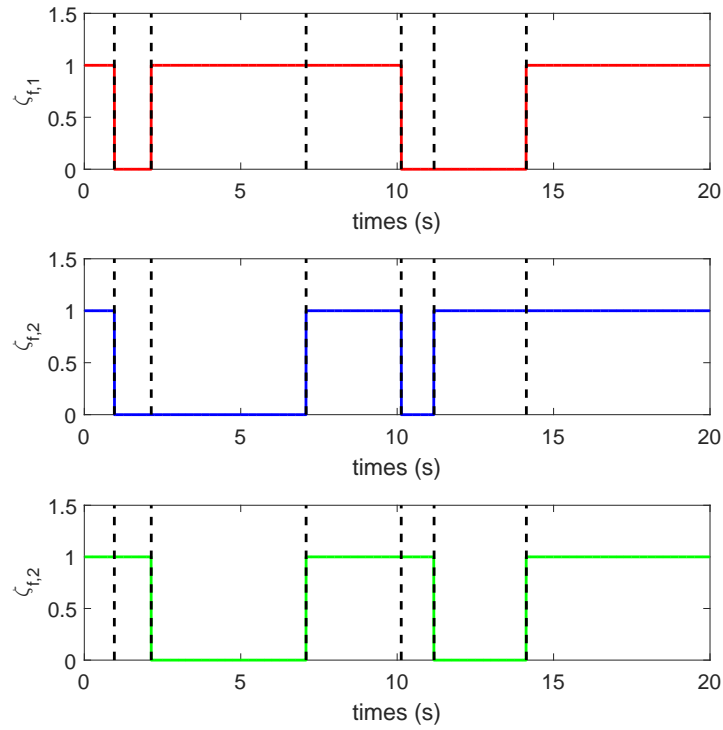


Figure 5.5: Leader-Follower relation for cooperative manipulators (No of agents: 2).

in these figures, this work can conclude that agents can avoid unknown obstacle effectively by adjusting the leader-follower relations.



(a) Avoidance results



(b) Histories of $\zeta_{f,i}$

Figure 5.6: Leader-Follower relation for cooperative manipulators (No of agents: 3).

6

Experimental Validation and Results

In this section, the experimental results are described with two custom-made aerial manipulators to carry an unknown payload. In the experiment, each aerial manipulator consists of a hexacopter and a 2-DOF arm. Two experimental results with two different types of payloads suggest that the proposed approach can be utilized for safe cooperative aerial transportation.

6.1 Simulation Validation for Moving Obstacle

Before validating the proposed method through experiments, simulation results are shown to analyze the effect of moving obstacles. Fig. 6.1 shows snapshot of the simulation during 10 seconds. Two aerial manipulator systems (AMS) are used. Here, the cyan and magenta solid line means the travelled trajectory of unknown obstacles. Fig. 6.2 presents position and attitude histories of the end-effector and the origin of the object. Fig. 6.3 addresses the estimated mass during flights. From the simulation results in Fig. 6.1-6.3, the proposed algorithm shows satisfactory tracking and estimation performance while transporting an unknown payload in unknown environments with moving obstacles.

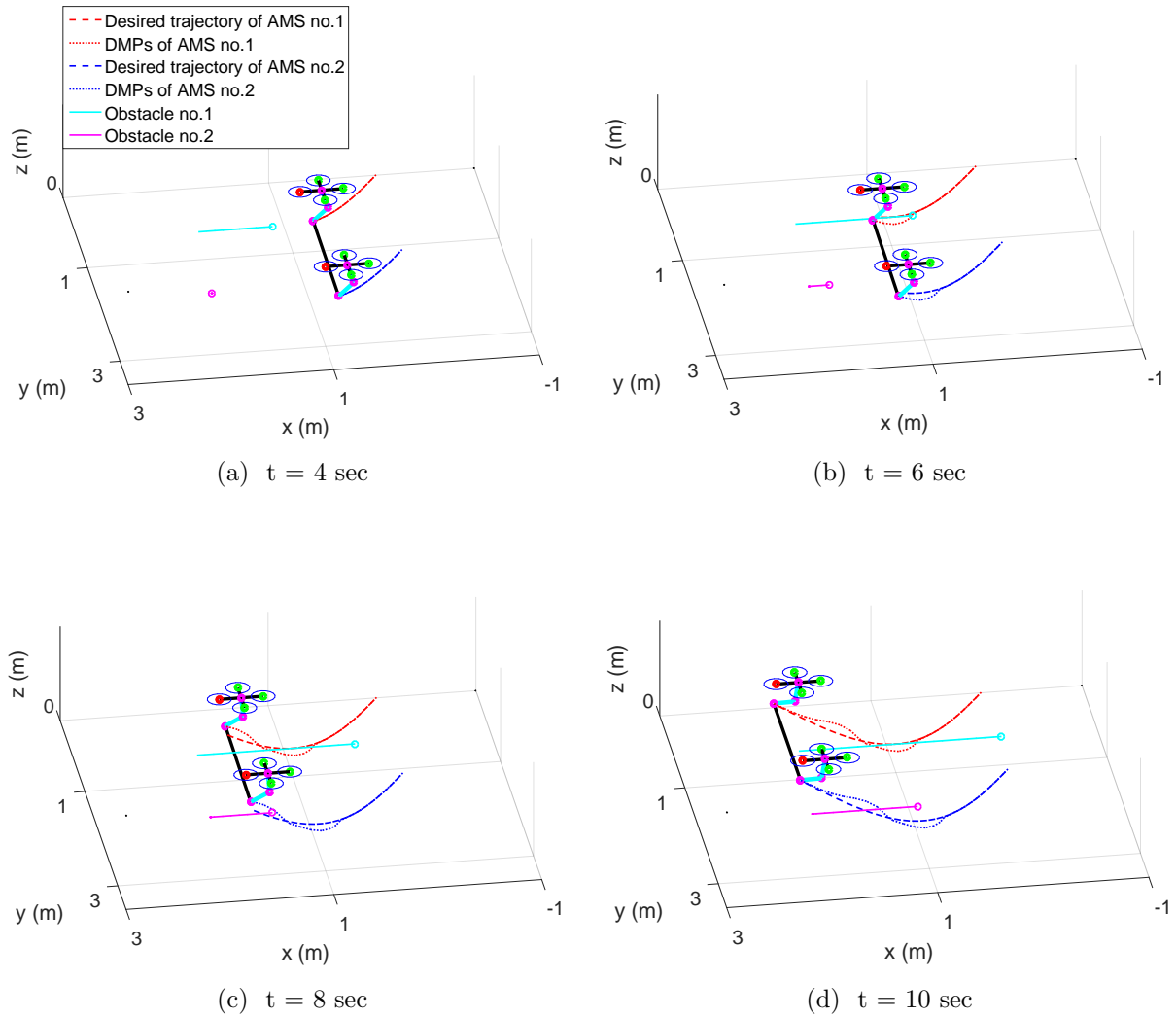
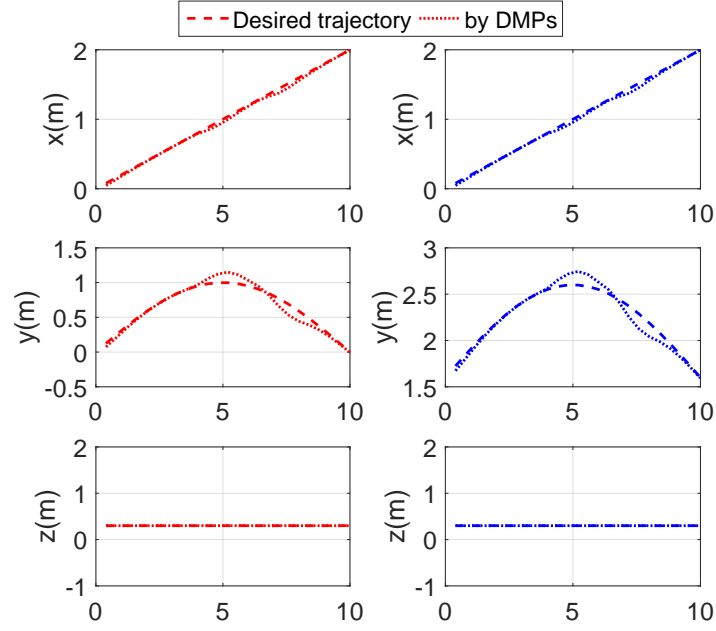
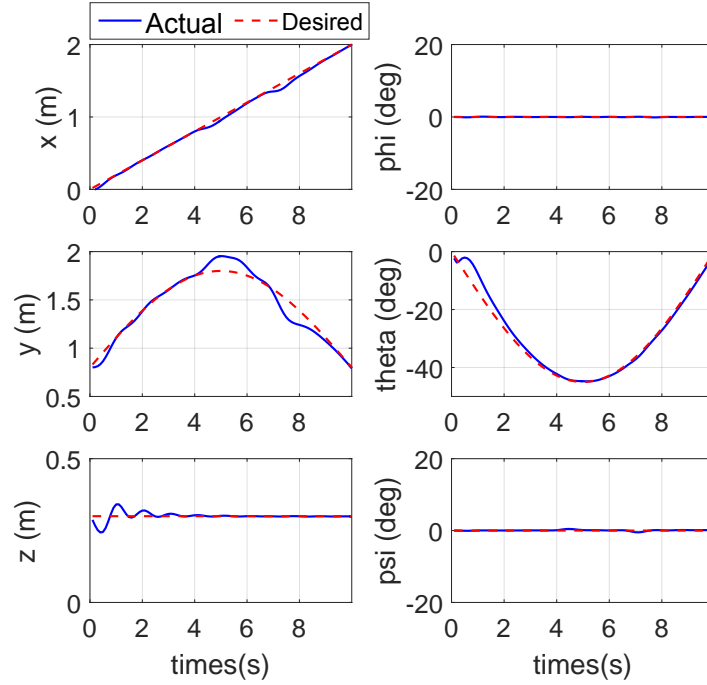


Figure 6.1: Snapshot of the simulation during 10 seconds. Two aerial manipulator systems (AMS) are used.



(a) Tracking performance of each end-effectors



(b) Time histories of \mathbf{q}_o

Figure 6.2: Position and attitude histories of the end-effector (a) and the object (b).

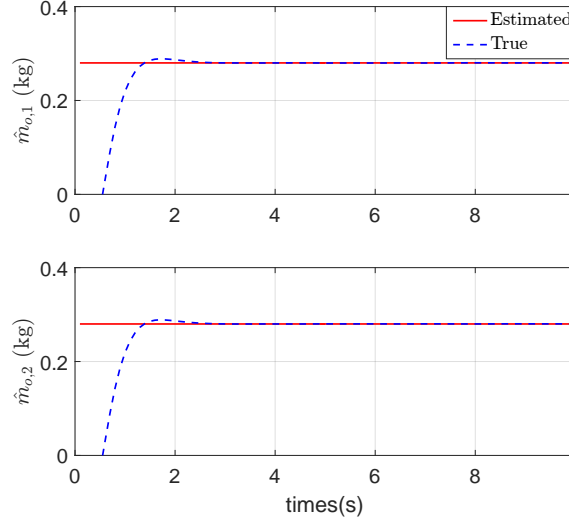
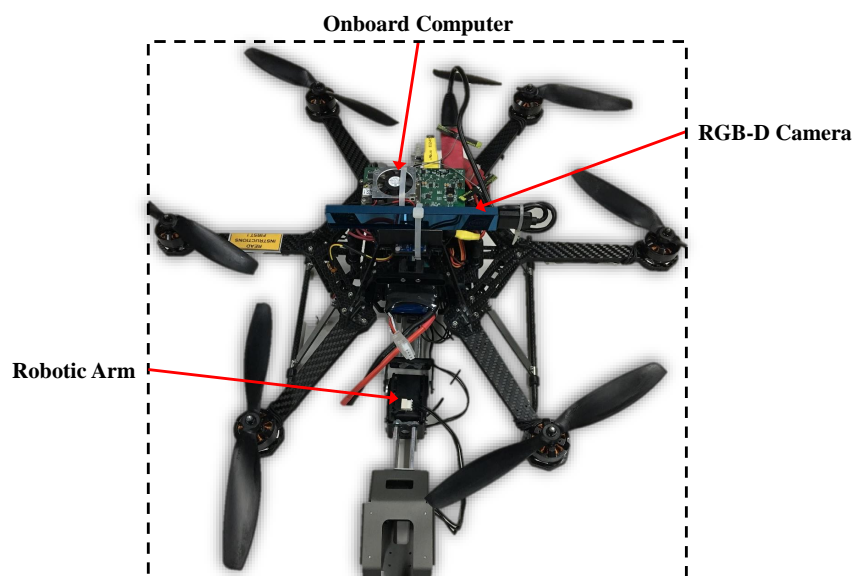


Figure 6.3: Estimated mass.

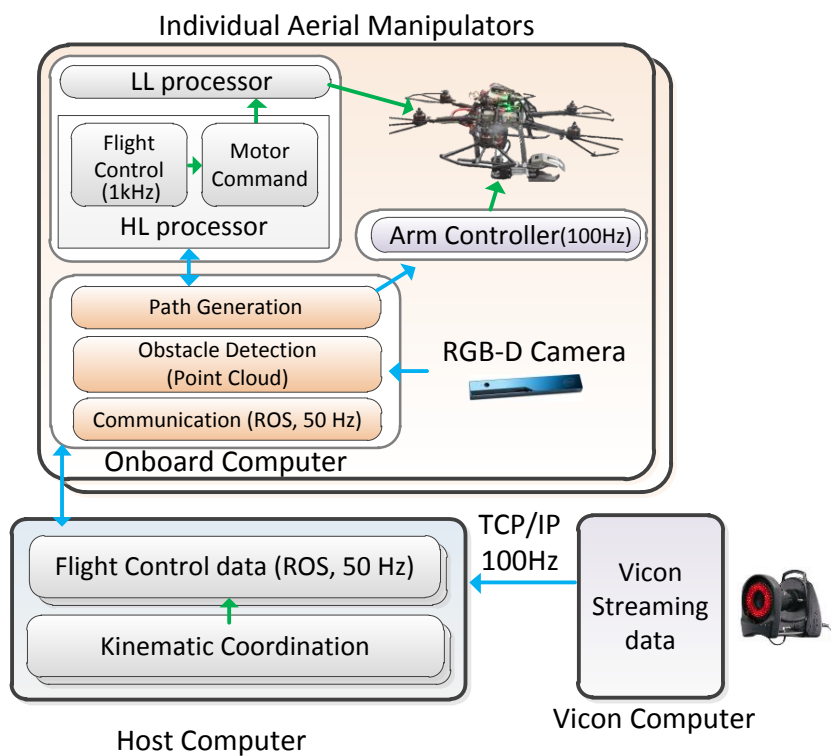
6.2 Experimental Setup

The platforms used in this dissertation are the Ascending Technologies Firefly hexacopters and each equipped with the 2 DOF robotic arm and RealSense RGB-D camera that provides 640×480 depth images. The robotic arms are customized with Dynamixel servomotors. The total length of each arm is 0.25 meter (i.e., $l_1 = l_2 = 0.125$). The total weight of the robotic arm is about 300 gram including the gripper before picking up the payload. The unknown payloads used in the experiment are a wooden rod of ‘I’ shape (1.8 meter, 280 gram) and a wooden rod of ‘L’ shape (0.9 meter for each side, 350 gram).

For the experiment, we use Vicon, an indoor GPS system, which measures the position information with 100 Hz as shown in Fig. 6.4. The implementation of the obstacle detection and the computation of the control inputs has been performed using Robot Operating Systems (ROS) [73], OpenCV and PCL. The OpenCV and PCL library include all the necessary functions for the image processing and point cloud, whereas ROS simplifies communication and synchronization issues that appear when working with multiple robots. The depth reprojection and cooperative control calculation execute in a loop 50 Hz using



(a) An aerial manipulator (1.58 kg)



(b) Structure for experiments

Figure 6.4: Experimental setup.



Figure 6.5: Joule compute module [10].

an Intel Joule PC equipped with an Intel Atom T5700 of four cores and 4 GB of RAM (see Fig. 6.5). In this experiment, to reduce the computational load for depth processing, depth images are downsampled as 160×120 . The desired trajectory of the aerial manipulator and actual states of hexacopters with respect to the inertial frame are transmitted to the hexacopter with Wifi. The proposed estimator and controller run at 1 kHz in the HL (High Level) processor of the hexacopters. The data between PC and HL processor are transmitted by serial communication. Note that communication delay between multiple robots or packet drop has not been considered in this experiment. The more details about time delay of networked robots can be found in [33].

The gain matrices are set as

$$K_s = \text{diag}[9.5, 9.5, 5.5, 1.0, 1.0, 1.0, 0.5, 0.5]$$

$$\Lambda_i = \text{diag}[0.35, 0.35, 4.0, 3.2, 3.2, 3.2, 1.0, 1.0]$$

$$K_\Delta = \text{diag}[2.0, 2.0, 3.0, 0.5, 0.5, 0.5, 0.0, 0.0]$$

For the parameter update, we set $\Gamma_1 = 0.05 \times I_{8 \times 8}$, $\Gamma_2 = 0.05 \times I_{8 \times 8}$, $C^* = 10 \times I_{8 \times 8}$ and $K^* = 20 \times I_{8 \times 8}$. The deactivation buffer is set to be $b_p = 0.05$ m. In addition for simplicity, we have assumed that translational and angular velocities are small, which means that their product is negligible, i.e., $C(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i \approx 0$.

For the desired trajectory, since the behavior of the end-effector of the aerial manipulator is highly affected by the attitude of the hexacopter, the desired trajectory can be oscillated

due to disturbances, which causes the performance degradation. To prevent this problem, we performed smoothing by applying the low-pass filter to the generated task solution $\dot{\nu}_{i,3}$ as appeared in (4.19).

6.3 Experiment for Cooperative Aerial Transportation

In this section, the results of autonomous transportation of two payloads are shown. This work performs two experiment: 1) path following with two types of payload and 2) aerial transportation in unknown environments. Through the experiments, this work validates the performance of the proposed algorithm on custom-made aerial manipulators with two types of unknown payloads.

6.3.1 Path Following with Two Types of Payloads

For the first experiment, cooperative aerial manipulators carry two type of wooden rods (i.e., ‘L’ shape and ‘I’ shape). For the rod of ‘L’ shape, cooperative aerial manipulators follow a straight line for the payload, while the rod of ‘I’ shape follows the circular trajectory.

The wooden rod of ‘L’ shape is carried by cooperative aerial manipulators. Fig. 6.6 shows the picture taken during the experiment. When cooperative aerial manipulators take off with the payload before tracking the straight line, they cannot determine the allowable flight region because they do not know the mass of the payload. For this reason, at first, they estimate the unknown mass in hover flight during the first 25 seconds. Then, they follow the desired trajectory of the end-effector during the next 25 seconds. Fig. 6.7 shows the time histories of \mathbf{q}_o . In this figure, the red line is the desired trajectory for the payload and blue line means the actual states. As demonstrated in our experiment, the proposed algorithm shows satisfactory tracking performance when handling an unknown payload. The parameter estimation results are shown in Fig. 6.8. By exploiting consensus algorithm, the parameter estimation error between each robot decreases as time goes by.

Fig. 6.9 shows the picture taken during the experiment for the rod of ‘I’ shape. Co-

operative aerial manipulators estimate the unknown mass in hover flight during the first 25 seconds. Then, they follow the circular trajectory for the payload during the next 37 seconds. Fig. 6.10a shows the time histories of \mathbf{q}_o for ‘I’-shape payload. The RMS (Root Mean Square) errors in x_o , y_o and z_o directions are described in table 6.1. Fig. 6.10b shows parameter estimation results with or with consensus algorithm. From these figures, this work prove that our proposed algorithm with consensus shows more satisfactory estimation results, while the estimation result without consensus has a clear error for the mass of the unknown common payload. In addition, since the consensus algorithm makes the estimation result of two manipulators identical, two aerial manipulators can share the same flight envelope, which results in the synchronized the motion of the robotic arms.

Table 6.1: The RMS errors of the ‘I’ type payload for circular trajectory tracking

	x_o	y_o	z_o
[m]	0.1475	0.2448	0.01917

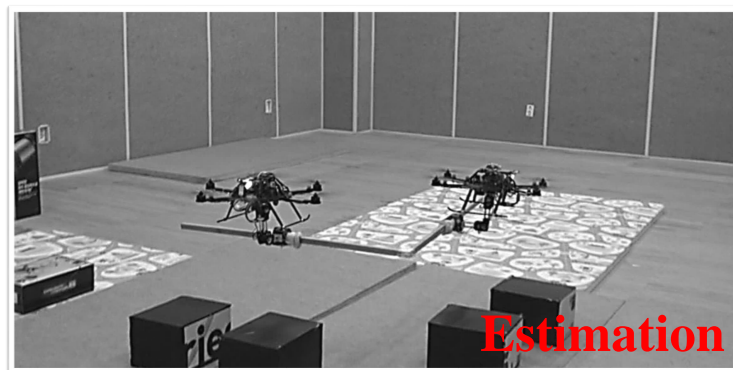
6.3.2 Aerial Transportations in Unknown Environments

For the second experiment, cooperative aerial manipulators transport the wooden rod of ‘I’ shape in unknown environments. In the environments, there is one obstacle near the desired path of an aerial manipulator as shown in Fig. 1.4.

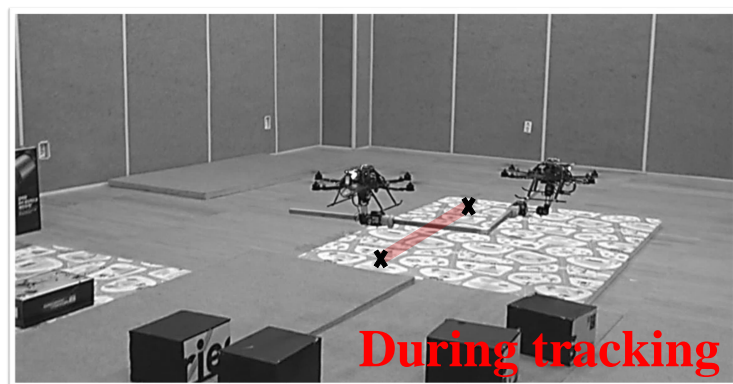
In order to illustrate the performance in more detail, we prepare flight scenarios to transport the object. The desired trajectory of the object is set to be:

$$\begin{aligned} \mathbf{p}_o^d &= \begin{bmatrix} 0.1 \times t & \frac{1}{2}(\cos(\frac{\pi}{30}t) - 1) & 0.45 \end{bmatrix} \\ \Phi_o^d &= \begin{bmatrix} 0 & 25 \sin(\frac{\pi}{40}t) - 25 & 0 \end{bmatrix}, \end{aligned} \quad (6.1)$$

where the unit of Φ_o^d is degree. In this experiment, we focus on the tracking performance of the proposed method while avoiding an unknown obstacle. To do so, this work shows



(a) Estimation phase ($t = 20$ s)



(b) Transporting an object ($t = 50$ s)

Figure 6.6: Pictures taken from the experiment for cooperative aerial transportation ('L' shape).

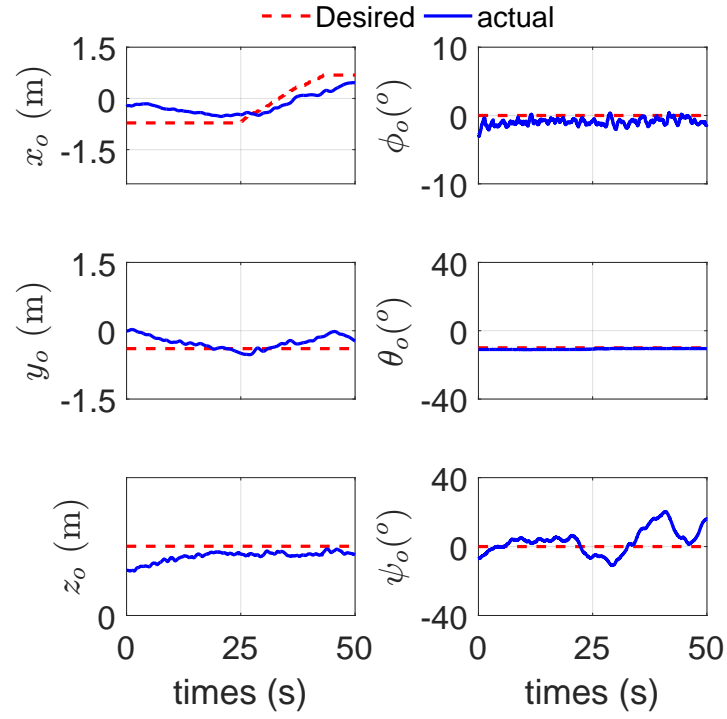


Figure 6.7: Time histories of \mathbf{q}_o ('L' shape rod).

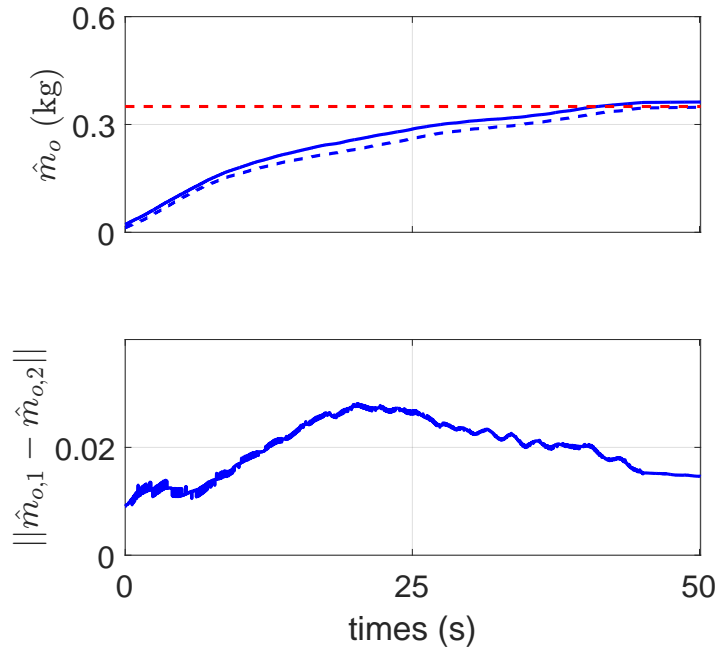
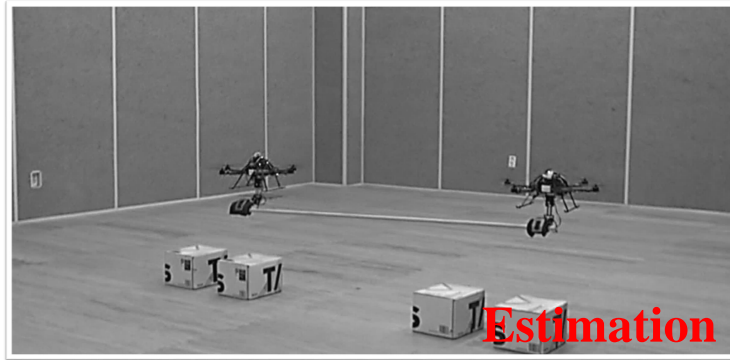
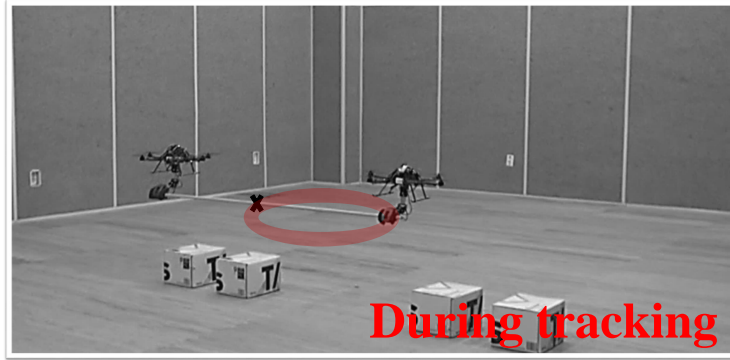


Figure 6.8: Parameter estimation ('L' shape rod).



(a) Estimation phase ($t = 20$ s)



(b) Transporting an object ($t = 50$ s)

Figure 6.9: Pictures taken from the experiment for cooperative aerial transportation ('T' shape).

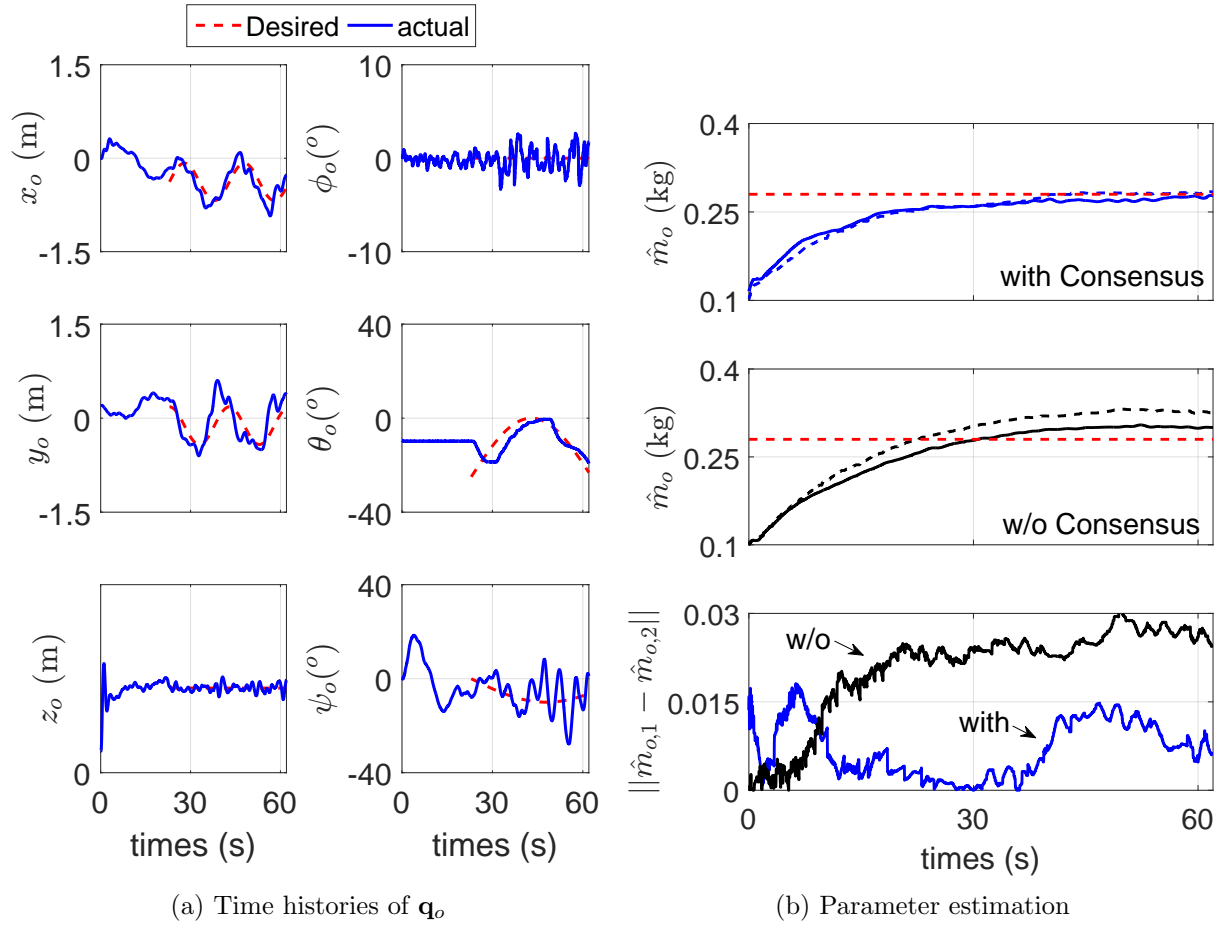


Figure 6.10: Estimation and tracking performance ('I' shape rod).

the results in two aspects: 1) trajectory tracking performance in the task and joint spaces and 2) the tracking error with or with rigidity maintenance algorithm by DMPs. First, for the tracking performance, this dissertation shows estimation results and the time histories of the desired and actual states. Second, the tracking performance with or with rigidity maintenance is compared. If not handling the rigidity between robots, the internal stability may cause a problem during avoiding an obstacle. Therefore, this work considers that the avoidance algorithm with or without the maintenance. In the case of ‘without rigidity maintenance’, the follower robot just tracks the original trajectory because the follower does not know the actual states of the leader robot.

Fig. 6.11 shows pictures taken during the flight experiment. As same with the previous experiments in Figs. 6.6 and 6.9, the cooperative aerial manipulators first estimate the physical properties of the payload and transport the payload. Fig. 6.12 shows time histories of \mathbf{q}_o . In this figure, the red dashed line means the desired state and blue line means the actual state of the cooperative robots. The RMS errors of the payload during trajectory tracking are appeared in table 6.2.

Table 6.2: The RMS errors for the payload tracking performance

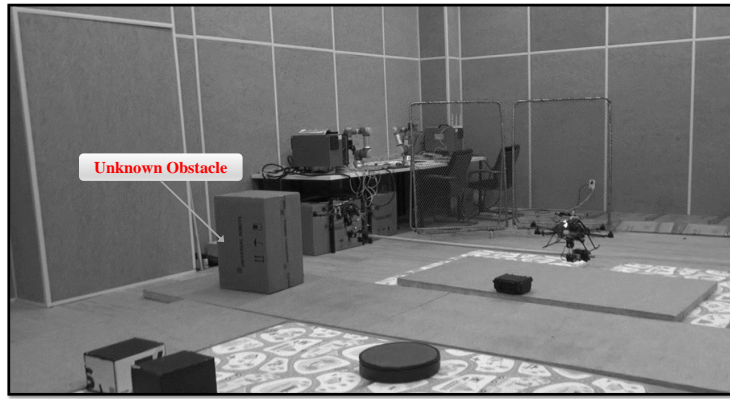
	x_o	y_o	z_o	ψ_o
[m, °]	0.1292	0.0902	0.0118	3.5383

Fig. 6.13 shows the results of parameter estimation. In this figure, the red dashed line means the unknown mass of the payload, the blue line means the parameter estimated by the first aerial manipulator and blue dashed line means the parameter estimated by the second aerial manipulator. Recalling that the true mass of the payload is 0.25 kg, the estimated parameter converged to the true value satisfactorily without using any force/torque sensors. In addition, exploiting the consensus rule, the difference between two estimated parameters also converges to zero almost.

Fig. 6.14 represents the relative distance with respect to the base frame $\Sigma_{b,i}$ and time histories of the common payload while avoiding an unknown obstacle. Fig. 6.15 shows



(a) Estimation phase ($t = 15$ s)



(b) Avoiding an unknown obstacle ($t = 30$ s)



(c) Arrival to the target position ($t = 50$ s)

Figure 6.11: Pictures taken from the experiment for cooperative aerial transportation.

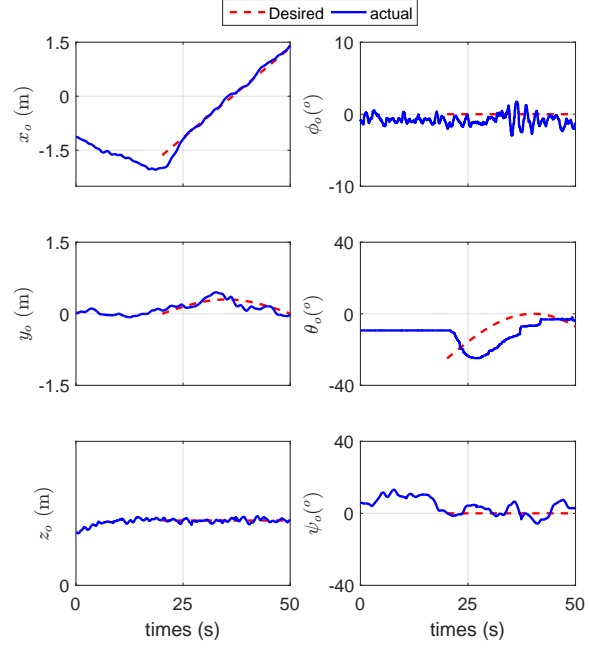


Figure 6.12: Time histories of \mathbf{q}_o ('I' shape rod).

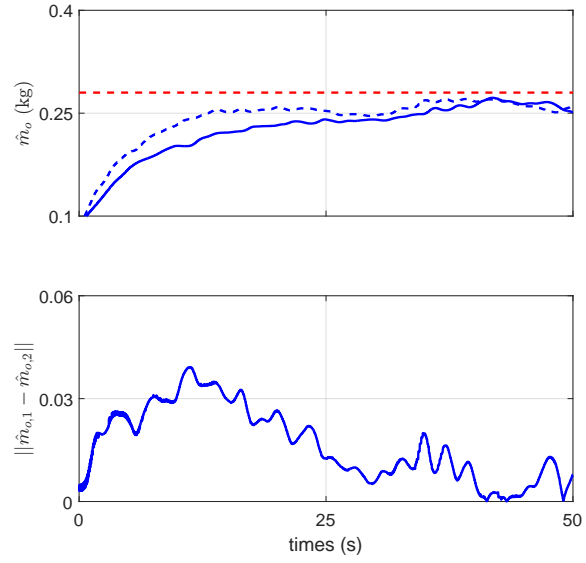
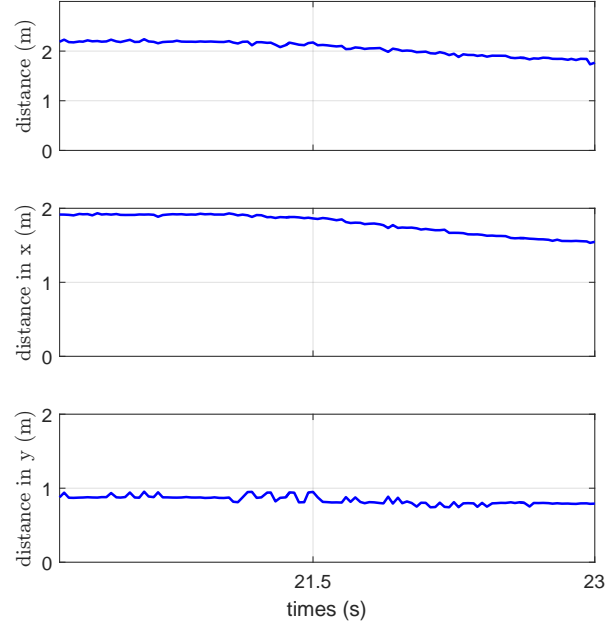
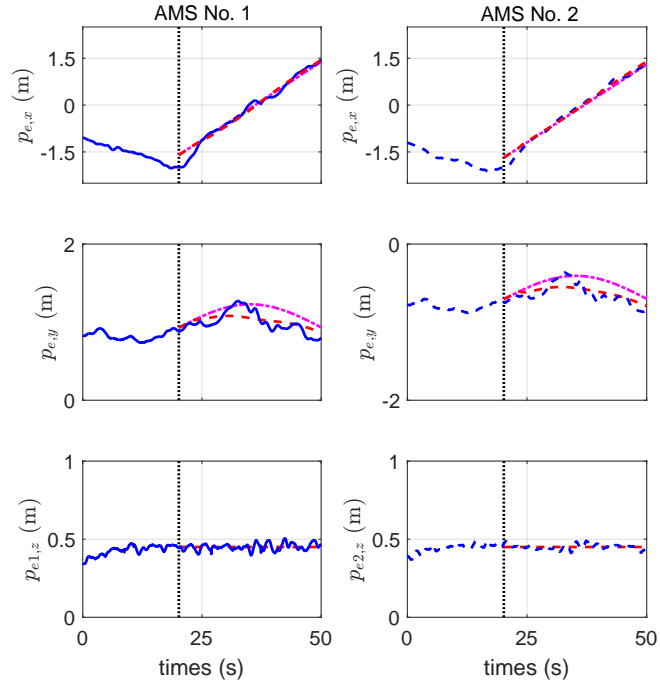


Figure 6.13: Parameter estimation ('I' shape rod).



(a) Relative distance between the leader and detected obstacle



(b) Time histories of the common object

Figure 6.14: Time histories of each end-effector with rigidity maintenance while avoiding an obstacle.

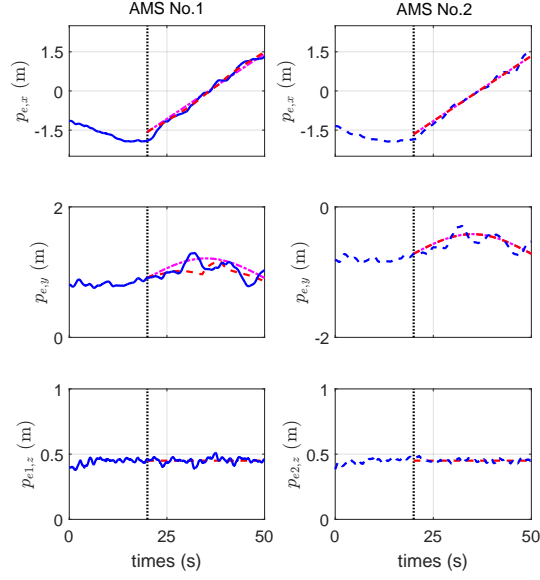


Figure 6.15: Time histories of each end-effector without rigidity maintenance (The proposed method).

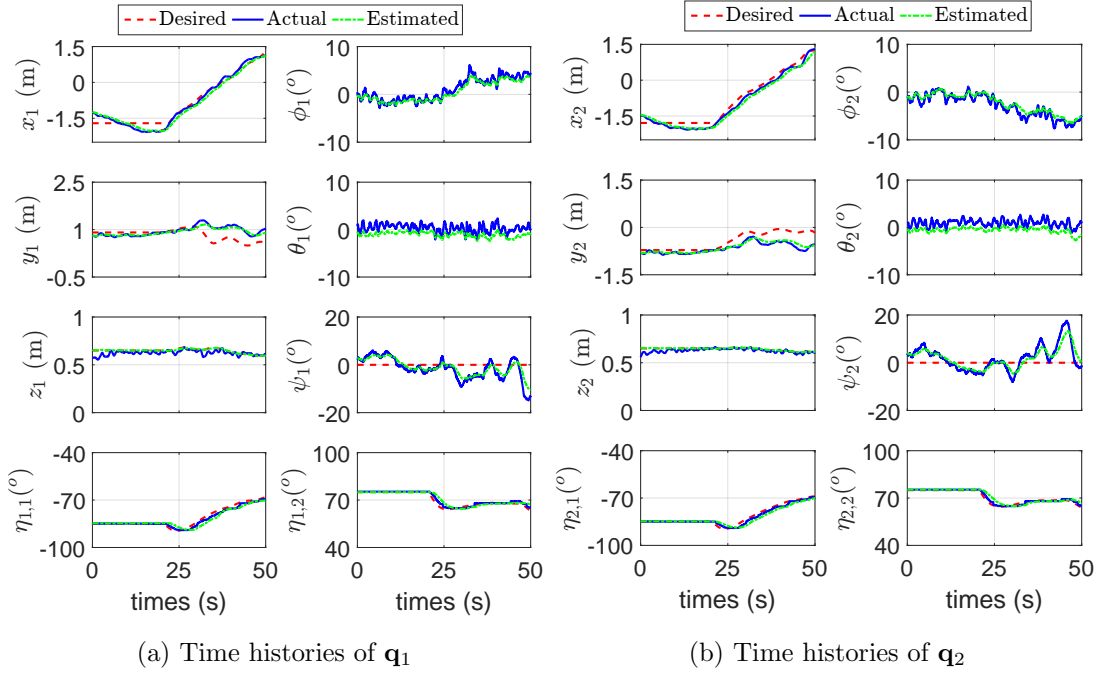


Figure 6.16: Time histories of \mathbf{q}_1 and \mathbf{q}_2 without rigidity maintenance.

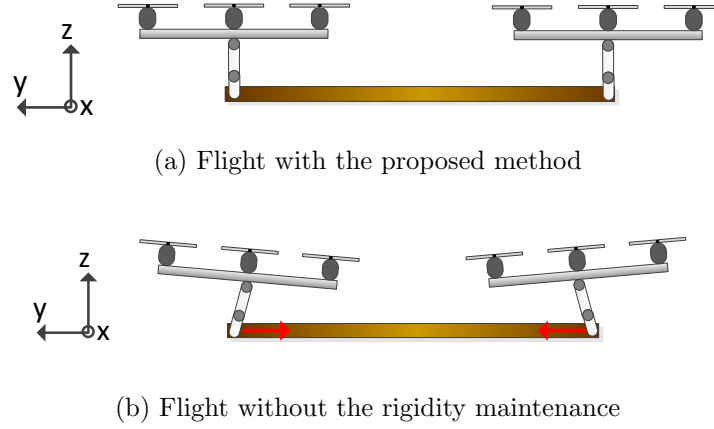


Figure 6.17: Configuration of the cooperative aerial manipulators while avoiding an obstacle.

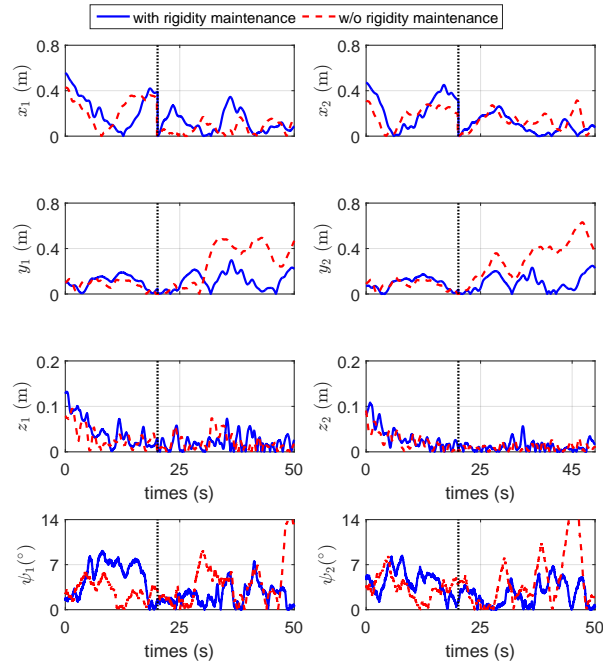


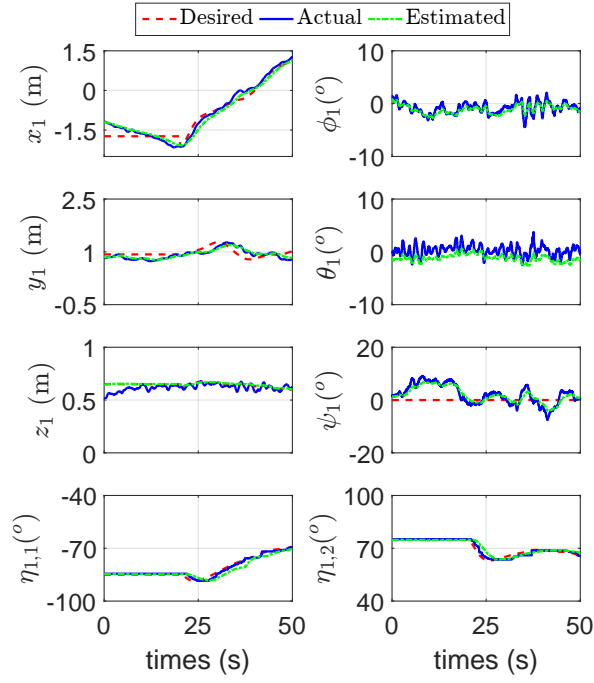
Figure 6.18: Tracking errors with or without rigidity maintenance.

Table 6.3: The RMS errors with or without rigidity maintenance algorithm

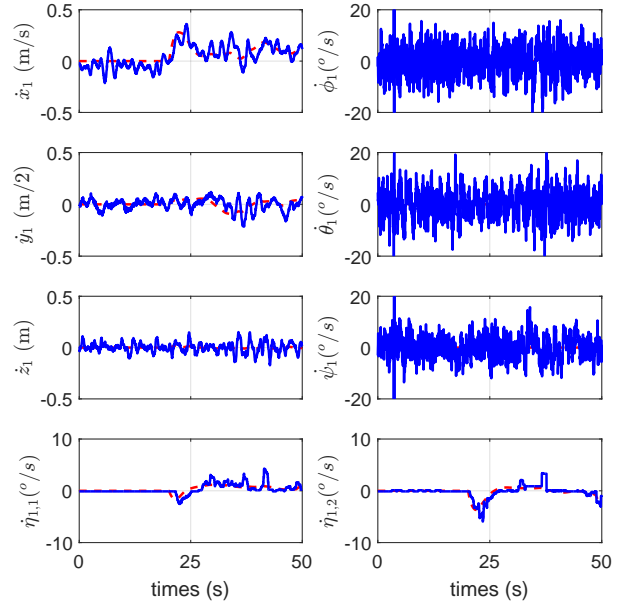
	x_1	y_1	z_1	ψ_1	x_2	y_2	z_2	ψ_2
with [m, °]	0.1552	0.1365	0.0261	2.7472	0.1281	0.1085	0.0147	2.9883
without [m, °]	0.0838	0.3250	0.0256	5.4369	0.1492	0.3511	0.0114	6.4076

time histories of each end-effector without rigidity maintenance. In these figures, the black dotted line means the starting point to track the desired trajectory after finishing the estimation. The magenta dotted line means the previously designed trajectory before avoiding an obstacle. The results in Figs. 6.14 and 6.15 show that the leader robot which detects an obstacle suitably modifies the desired trajectory to avoid an obstacle and the follower robot also modifies the trajectory. Fig. 6.16 show the time histories of \mathbf{q}_1 and \mathbf{q}_2 to track the corresponding trajectory of the end-effector without rigidity maintenance. From this figure, the roll and yaw angles are unambiguously larger than the result in Fig. 6.19. So, it is possible to say that the proposed method can reduce the internal forces while avoiding an obstacle as described in Fig. 6.17. Therefore, the proposed method presents more stable than the method without the rigidity maintenance algorithm. In fig. 6.18, tracking errors are shown with or without cases. In this experiment, since the desired yaw is zero while moving forward and transporting a payload, the tracking errors can be larger in y and yaw direction than in x, z direction. The RMS errors is appeared in table 6.3. This table verifies that the tracking errors without rigidity maintenance are much larger than the errors with the proposed method. This means that the system without maintenance algorithm is more unstable than the system with the maintenance.

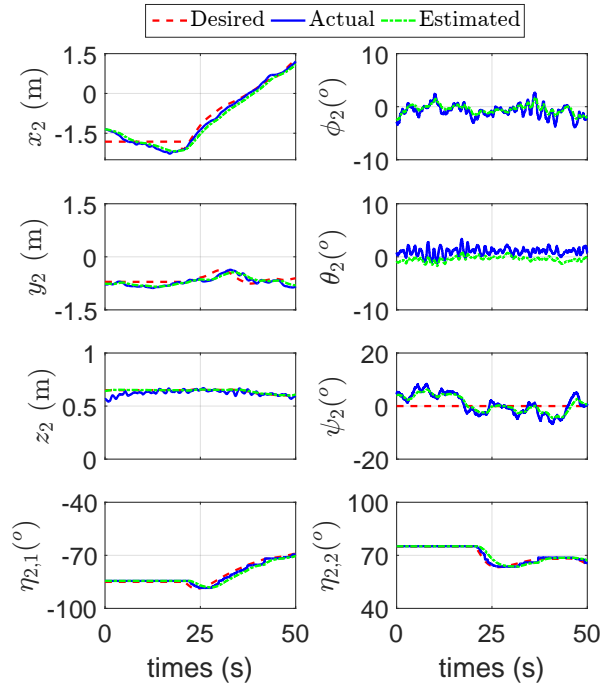
Fig. 6.19 shows the time histories of \mathbf{q}_i and $\dot{\mathbf{q}}_i$ for the proposed method by using the task priority solution. In fig. 6.20, attitude tracking of \mathbf{q}_o and constraints for \mathbf{q}_i are shown. The magenta dashed lines in this figure mean the unilateral constraints for the end-effector to satisfy the allowable flight envelope. Although the constraints can vary depending on the estimated mass, both constraints $\bar{x}_{e,1}$ and $\bar{x}_{e,2}$ are set to be almost the same thanks to the consensus algorithm. Based on our proposed synthesis, it can be said that the aerial



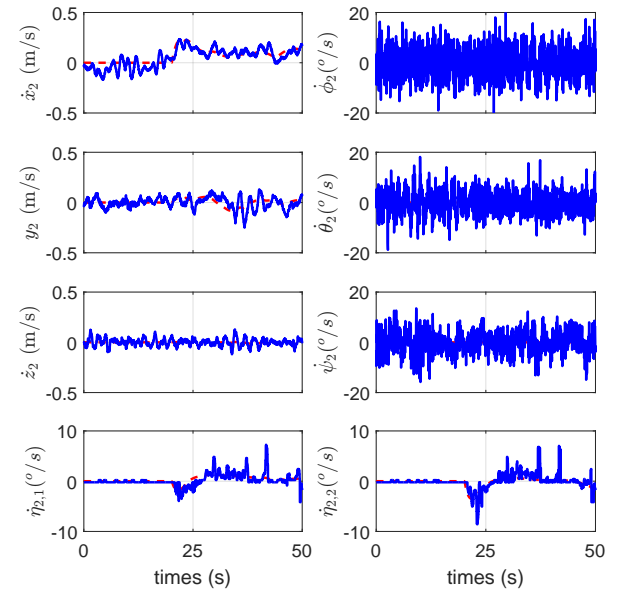
(a) Time histories of \mathbf{q}_1



(b) Time histories of $\dot{\mathbf{q}}_1$

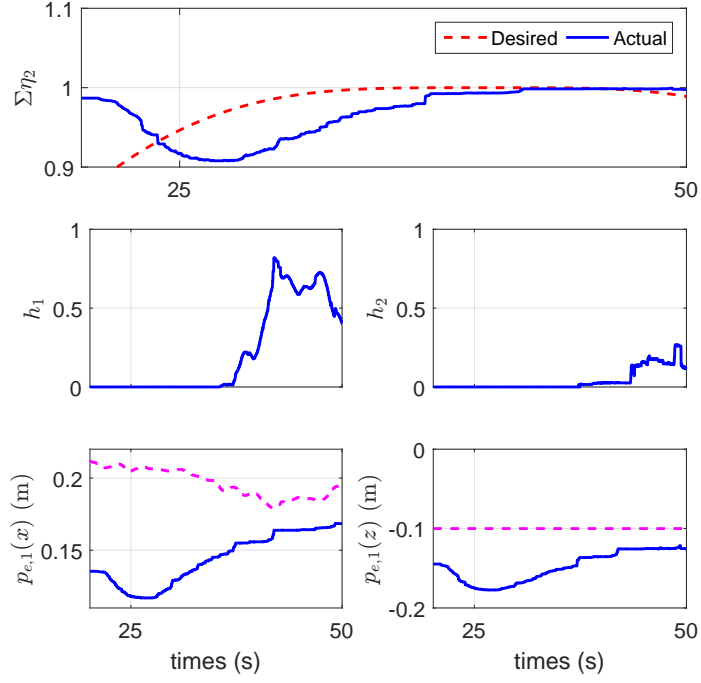


(c) Time histories of $\dot{\mathbf{q}}_2$

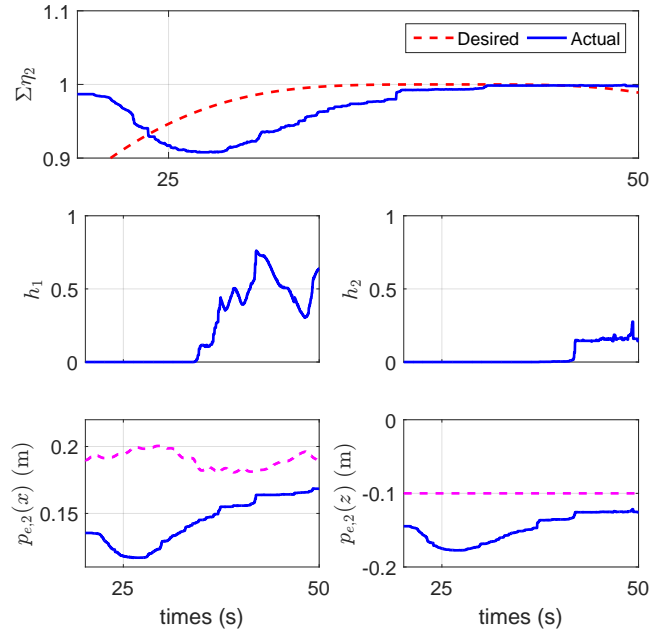


(d) Time histories of $\dot{\mathbf{q}}_2$

Figure 6.19: Time histories of \mathbf{q}_i and $\dot{\mathbf{q}}_i$.



(a) Constraints of \mathbf{q}_1



(b) Constraints of \mathbf{q}_2

Figure 6.20: Attitude tracking of \mathbf{q}_o and constraints for \mathbf{q}_i .

manipulators are possible to transport the unknown payload safely while satisfying the unilateral constraints in $\bar{x}_{e,1}$ and $\bar{z}_{e,1}$.

7

Conclusions

This dissertation studies on estimation, control and planning synthesis for cooperative aerial transportation while avoiding an unknown obstacle. The whole study is divided into three topics: (i) estimation and control of cooperative aerial manipulators, (ii) motion planning within the allowable flight envelope and (iii) real-time obstacle avoidance by using RGB-D camera and DMPs. By exploiting three topics, this work shows two experimental results: (i) trajectory tracking with two different types of the payload and (i) obstacle avoidance in the unknown environments.

- First topic was the estimation and control synthesis. In order to design the parameter estimator and the controller, this work addressed dynamics of the cooperative aerial manipulators. A closed-chain dynamics consisting of multiple aerial manipulators and a common payload was presented with the rigid gripper. In addition, from the assumption that the common object is rigid and the mass is uniformly distributed over the whole object, the decoupled dynamics for the combined system of the i th aerial manipulator was obtained in joint space \mathbf{q}_i . Based on the parameterized model of the decoupled dynamics, this work designed an on-line parameter estimator only using

the measurement of states of the system, not using heavy or expensive force/torque sensors. An augmented adaptive sliding mode controller was designed to handle modeling uncertainties or grasp error. To handle the none-skew-symmetric property of the decoupled dynamics, the proposed controller contained the feed-forward terms to make the system stable. Finally, by exploiting Lyapunov stability theorem, this work proved that the proposed estimation and control synthesis is stable.

- In the second part, this work studied a motion planning algorithm for the cooperative aerial manipulators. To perform the transportation mission, this dissertation proposed a trajectory generation by using the task priority solution. For the safe aerial transportation, the allowable flight envelope was proposed in the consideration of the position of the end-effector and the limit of motors on the aerial robot. From the allowable flight envelope, the unilateral constraints were obtained for the stable flight. To satisfy the unilateral constraints, the task priority solution was used for each aerial manipulator. The main trajectory for transportation was derived based on the analysis of the kinematic relationship between an aerial robot and the corresponding end-effector. From the experiment with a single aerial manipulator, this work addressed that the proposed trajectory generation method could be more applicable for an aerial manipulator with higher DOF of the robotic arm in comparison with the conventional null space approach.
- The third topic was the detection and avoidance of an unknown obstacle. In order to avoid an unknown obstacle for the cooperative aerial manipulator, this dissertation proposed the avoidance algorithm with DMPs and RGB-D camera. To increase the performance of obstacle detection, the voxel-grid filter and statistical filter were used. These filters could remove the measurement noise in the depth image. After reconstructing 3D point cloud map, an obstacle was recognized by the closest point. To avoid an unknown obstacle, DMPs was employed for the cooperative aerial manipulators with the leader-follower approach. The leader robot, which detects an obstacle

first, modified its original trajectory by using a potential function. The other follower robots also modified their trajectory to maintain the rigidity constraints with the leader.

- Finally, this dissertation showed two experimental results by using custom-made aerial manipulators. The aerial manipulator consisted of an onboard computer and an RGB-D camera. To validate the proposed method, this work presented two experimental results. In the first experiment, trajectory tracking experiment with two different types of payloads was shown. In the second experiment, the cooperative aerial manipulators avoided an unknown obstacle while transporting a payload. From these two experiments, this work suggested that the proposed approach could be utilized for safe cooperative aerial transportation.

References

- [1] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, “Cooperative grasping and transport using multiple quadrotors,” in *Distributed autonomous robotic systems*. Springer, 2013, pp. 545–558.
- [2] N. Michael, J. Fink, and V. Kumar, “Cooperative manipulation and transportation with aerial robots,” *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.
- [3] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo, “The flying hand: A formation of uavs for cooperative aerial tele-manipulation,” in *IEEE International Conference on Robotics and Automation*, 2014, pp. 4335–4341.
- [4] H. Yang and D. Lee, “Hierarchical cooperative control framework of multiple quadrotor-manipulator systems,” in *IEEE International Conference on Robotics and Automation*, 2015, pp. 4656–4662.
- [5] G. Muscio, F. Pierri, M. Trujillo, E. Cataldi, G. Giglio, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, “Experiments on coordinated motion of aerial robotic manipulators,” in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1224–1229.
- [6] H. Lee and H. J. Kim, “Estimation, control, and planning for autonomous aerial transportation,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3369–3379, April 2017.
- [7] S. Erhart and S. Hirche, “Model and analysis of the interaction dynamics in cooperative manipulation tasks,” *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 672–683, June 2016.

- [8] Y. Kume, Y. Hirata, and K. Kosuge, “Coordinated motion control of multiple mobile manipulators handling a single object without using force/torque sensors,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 4077–4082.
- [9] A. Perez, S. Karaman, A. Shkolnik, E. Frazzoli, S. Teller, and M. R. Walter, “Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 4307–4313.
- [10] “Discover the intel joule compute module,” <https://software.intel.com/en-us/iot/hardware/joule>.
- [11] H. Lee and H. J. Kim, “Trajectory tracking control of multirotors from modelling to experiments: A survey,” *International Journal of Control, Automation and Systems*, vol. 15, no. 1, pp. 281–292, 2017.
- [12] J. Fink, N. Michael, S. Kim, and V. Kumar, “Planning and control for cooperative manipulation and transportation with aerial robots,” *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 324–334, 2011.
- [13] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, “Cooperative impedance control for multiple uavs with a robotic arm,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 2366–2371.
- [14] H. Lee, H. Kim, and H. J. Kim, “Planning and control for collision-free cooperative aerial transportation,” *IEEE Transactions on Automation Science and Engineering*, Accepted for publication.
- [15] —, “Path planning and control of multiple aerial manipulators for a cooperative transportation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2015, pp. 2386–2391.

- [16] D. Devaurs, T. Siméon, and J. Cortés, “Optimal path planning in complex cost spaces with sampling-based algorithms,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 415–424, 2016.
- [17] A. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, “Control of an aerial robot with multi-link arm for assembly tasks,” in *IEEE International Conference on Robotics and Automation*, 2013.
- [18] G. Garimella and M. Kobilarov, “Towards model-predictive control for aerial pick-and-place,” in *IEEE International Conference on Robotics and Automation*, 2015, pp. 4692–4697.
- [19] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, “Modeling and control of a flying robot for contact inspection,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3532–3537.
- [20] T. Yoshikawa and X.-Z. Zheng, “Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object,” *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 219–230, 1993.
- [21] D. Sieber, F. Deroo, and S. Hirche, “Formation-based approach for multi-robot cooperative manipulation based on optimal control design,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5227–5233.
- [22] M. Gouttefarde, J. Lamaury, C. Reichert, and T. Bruckmann, “A versatile tension distribution algorithm for n -dof parallel robots driven by $n + 2$ cables,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1444–1457, 2015.
- [23] M. Gouttefarde, J.-F. Collard, N. Riehl, and C. Baradat, “Geometry selection of a redundantly actuated cable-suspended parallel robot,” *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 501–510, 2015.

- [24] P. Chiacchio, S. Chiaverini, and B. Siciliano, “Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system,” *Journal of dynamic systems, measurement, and control*, vol. 118, no. 4, pp. 691–697, 1996.
- [25] F. Caccavale, P. Chiacchio, and S. Chiaverini, “Task-space regulation of cooperative manipulators,” *Automatica*, vol. 36, no. 6, pp. 879–887, 2000.
- [26] J. J. Craig, P. Hsu, and S. S. Sastry, “Adaptive control of mechanical manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 16–28, 1987.
- [27] W. Dong, “On trajectory and force tracking control of constrained mobile manipulators with parameter uncertainty,” *Automatica*, vol. 38, no. 9, pp. 1475–1484, 2002.
- [28] B. Xiao, S. Yin, and O. Kaynak, “Tracking control of robotic manipulators with uncertain kinematics and dynamics,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6439–6449, 2016.
- [29] F. Aghili, “Adaptive control of manipulators forming closed kinematic chain with inaccurate kinematic model,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 5, pp. 1544–1554, 2013.
- [30] R. Monfaredi, S. M. Rezaei, and A. Talebi, “A new observer-based adaptive controller for cooperative handling of an unknown object,” *Robotica*, vol. 34, no. 07, pp. 1437–1463, 2016.
- [31] H. Kawasaki, S. Ueki, and S. Ito, “Decentralized adaptive coordinated control of multiple robot arms without using a force sensor,” *Automatica*, vol. 42, no. 3, pp. 481–488, 2006.
- [32] X. Zhao, C. Ma, X. Xing, and X. Zheng, “A stochastic sampling consensus protocol of networked euler-lagrange systems with application to two-link manipulator,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 907–914, Aug 2015.

- [33] X. Liang, H. Wang, Y. H. Liu, W. Chen, G. Hu, and J. Zhao, “Adaptive task-space cooperative tracking control of networked robotic manipulators without task-space velocity measurements,” *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2386–2398, Oct 2016.
- [34] A. Franchi, A. Petitti, and A. Rizzo, “Decentralized parameter estimation and observation for cooperative mobile manipulation of an unknown load using noisy measurements,” in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5517–5522.
- [35] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, “Design, modeling, estimation and control for aerial grasping and manipulation,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2668–2673.
- [36] H. Liu, X. Wang, and Y. Zhong, “Quaternion-based robust attitude control for uncertain robotic quadrotors,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 406–415, 2015.
- [37] P. E. Pounds, D. R. Bersak, and A. M. Dollar, “Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control,” *Autonomous Robots*, vol. 33, no. 1-2, pp. 129–142, 2012.
- [38] F. Morbidi, R. Cano, and D. Lara, “Minimum-energy path generation for a quadrotor uav,” in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1492–1498.
- [39] R. Rossi, A. Santamaria-Navarro, J. Andrade-Cetto, and P. Rocco, “Trajectory generation for unmanned aerial manipulators through quadratic programming,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 389–396, April 2017.

- [40] M. Korayem, M. Irani, A. Charesaz, A. Korayem, and A. Hashemi, “Trajectory planning of mobile manipulators using dynamic programming approach,” *Robotica*, vol. 31, no. 04, pp. 643–656, 2013.
- [41] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [42] D. Lau, J. Eden, and D. Oetomo, “Fluid motion planner for nonholonomic 3-d mobile robots with kinematic constraints,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1537–1547, 2015.
- [43] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, “Aerial robotic contact-based inspection: planning and control,” *Autonomous Robots*, pp. 1–25, Apr. 2015.
- [44] D. J. Webb and J. van den Berg, “Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics,” in *IEEE International Conference on Robotics and Automation*, May 2013, pp. 5054–5061.
- [45] A. Neto, D. G. Macharet, and M. F. Campos, “Feasible RRT-based path planning using seventh order bézier curves,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 1445–1450.
- [46] J. Gammell, S. Srinivasa, and T. Barfoot, “Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 2997–3004.
- [47] A. J. Ijspeert, J. Nakanishi, and S. Schaal, “Learning rhythmic movements by demonstration using nonlinear oscillators,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2002, pp. 958–963.
- [48] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, “Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle

- avoidance,” in *IEEE International Conference on Robotics and Automation*, May 2009, pp. 2587–2592.
- [49] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, “Coupling movement primitives: interaction with the environment and bimanual tasks,” *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 816–830, Aug. 2014.
 - [50] J. Umlauft, D. Sieber, and S. Hirche, “Dynamic movement primitives for cooperative manipulation and synchronized motions,” in *IEEE International Conference on Robotics and Automation*, May 2014, pp. 766–771.
 - [51] H. Lee and H. J. Kim, “Constraint-based cooperative control of multiple aerial manipulators for handling an unknown payload,” *IEEE Transactions on Industrial Informatics*, Accepted for publication, 2017.
 - [52] R. B. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1–4.
 - [53] A. Aldoma, Z. C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, “Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation,” *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 80–91, Sept 2012.
 - [54] S. Chiaverini, G. Oriolo, and I. D. Walker, “Kinematically redundant manipulators,” in *Springer handbook of robotics*. Springer, 2008, pp. 245–268.
 - [55] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.
 - [56] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

- [57] V. Lippiello and F. Ruggiero, “Exploiting redundancy in cartesian impedance control of uavs equipped with a robotic arm,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3768–3773.
- [58] M. C. Achtelik, K.-M. Doth, D. Gurdan, and J. Stumpf, “Design of a multi rotor mav with regard to efficiency, dynamics and redundancy,” in *AIAA Guidance, Navigation, and Control Conference*, 2012, pp. 1–17.
- [59] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, “Nonlinear robust adaptive tracking control of a quadrotor uav via immersion and invariance methodology,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 2891–2902, 2015.
- [60] S. Erhart and S. Hirche, “Internal force analysis and load distribution for cooperative multi-robot manipulation,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1238–1243, Oct 2015.
- [61] I. D. Walker, R. A. Freeman, and S. I. Marcus, “Analysis of motion and internal loading of objects grasped by multiple cooperating manipulators,” *The International journal of robotics research*, vol. 10, no. 4, pp. 396–409, 1991.
- [62] H.-C. Lin, T.-C. Lin, and K. Yae, “On the skew-symmetric property of the newton-euler formulation for open-chain robot manipulators,” in *American Control Conference*, vol. 3, 1995, pp. 2322–2326.
- [63] B. Siciliano and J.-J. E. Slotine, “A general framework for managing multiple tasks in highly redundant robotic systems,” in *International Conference on Advanced Robotics*, 1991, pp. 1211–1216.
- [64] F. Flacco and A. De Luca, “A reverse priority approach to multi-task control of redundant robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2421–2427.

- [65] A. Colomé and C. Torras, “Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements,” *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 944–955, 2015.
- [66] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: learning attractor models for motor behaviors,” *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [67] J. J. Lim, H. Pirsiavash, and A. Torralba, “Parsing ikea objects: Fine pose estimation,” in *IEEE International Conference on Computer Vision*, 2013, pp. 2992–2999.
- [68] H. K. Khalil and J. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, 2002, vol. 3.
- [69] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, “Task-priority based redundancy control of robot manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
- [70] F. Flacco and A. De Luca, “Unilateral constraints in the reverse priority redundancy resolution method,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 2564–2571.
- [71] N. Mansard, O. Khatib, and A. Kheddar, “A unified approach to integrate unilateral constraints in the stack of tasks,” *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.
- [72] S. Song, S. P. Lichtenberg, and J. Xiao, “Sun rgb-d: A rgb-d scene understanding benchmark suite,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 567–576.
- [73] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

국 문 초 록

최근들어 무인 드론을 활용한 비행 매니플레이터는 3차원 공간상의 뛰어난 기동성으로 인해 많은 주목을 받고 있다. 많은 연구진들은 비행 매니플레이터를 이용하여 사람의 손이 닿기 힘든 구조물로의 접근이나 무인 비행 운송등에 적용하기 위한 연구를 하고 있다. 본 박사 학위 논문에서는 복수의 비행 매니플레이터들을 이용한 미지 환경에서의 안전 비행운송을 위한 제어 및 경로 생성 기법을 제안하였다. 힘 분배 기법이나 임피던스 제어 등과 같이 힘/토크 센서를 필요로 하는 기존의 기법과는 달리, 본 학위논문에서는 협업 매니플레이터를 위한 비결합 운동방정식(Decoupled dynamics), 파라미터 추정기 및 제어기를 기반으로 하여 힘/토크 센서에 의존하지 않는 기법을 제안하였다. 이를 위해, 화물과 엔드이펙터가 강하게 결합되어 있다는 가정과 매개변수화된 모델(Parameterized model)을 기반으로, 비행 매니플레이터의 상태변수만을 이용하여 미지 물체의 물리량을 실시간으로 추정하는 파라미터 추정기가 제안되었다. 또한, 리아푸노프 안정성 이론을 바탕으로, 추정된 파라미터를 고려한 적응제어기가 설계 되었고, 증명을 통해 각 비행 매니플레이터들이 안정함을 보였다. 기존의 수동성 기반의(Passivity-based) 적응제어 기법과의 비교 시뮬레이션을 통해 본 학위논문에서 제안한 기법이 기존의 방법보다 더욱 만족스러운 결과를 보임을 나타내었다.

미지 화물을 안전하게 운송하기 위하여, 본 학위논문에서는 협업 비행 매니플레이터를 위한 경로생성 기법을 제안하였다. 비행 매니플레이터가 미지의 무거운 화물을 서랍장과 같이 자신의 팔을 뻗어야 하는 구조물로 운송하게 되는 경우, 무거운 물체로 인해 발생하는 추가적인 토크는 비행로봇 모터의 구동하게 범위를 넘게 할 수 있다. 이러한 문제점을 해결하기 위하여, 엔드 이펙터의 위치에 따른 비행 매니플레이터의 안전 운송 영역을 분석하였다. 엔드 이펙터가 비행 운송영역 안에 머무르게 하기 위하여, 우선, 드론과 엔드이펙터의 기구학(Kinematics)을 바탕으로 한 엔드 이펙터의 궤적생성부와 태스크 우선순위(Task priority) 기법을 활용하여 자신의 엔드 이펙터의 경로를 추종하기 위해 비행 매니플레이터의 경로를 생성하는 두 구조가 계층적으로 결합된 2단 경로생성 기법이 개발하였다. 비행 매니플레이터의 경로 생성시에는 비행 운송 영역으로 부터 습득된 엔드 이펙터의 구속조건을 고려한

경로가 생성되게 함으로써, 안전한 비행 운송이 가능하도록 하였다.

본 학위논문에서는 또한, 미지 환경에서의 비행운송을 위하여 협업 비행 매니플레이터의 장애물 회피 기법을 연구하였다. RGB-D 카메라와 PCL(Point Cloud Library)를 활용하여, 카메라와 장애물간의 상대적인 거리를 계산하고, 이를 바탕으로 사전에 계산된 경로를 장애물에 따라 수정하며 비행하도록 하는 장애물 회피기법을 제안하였다. 장애물을 회피하는 리더 로봇과 리더 로봇과의 거리를 유지하며 비행하는 팔로워 로봇의 구조를 적용함으로써, 복수 비행 매니플레이터의 장애물 회피 효율성을 높였다.

마지막으로, 본 학위논문에서는 자체 제작된 비행 매니플레이터를 이용하여 제안된 추정, 제어 및 경로 생성 기법을 실험을 통해 검증하였다. 제안된 기법의 확장 가능성을 위해 협업 비행 매니플레이터가 구조가 다른 두가지의 화물을 운송하는 경로 추정 실험과 미지 환경에서의 장애물 회피하는 실험을 수행하였다. 두 실험결과를 통해, 제안된 기법이 협업 비행 운송에 충분히 활용될 수 있다는 사실을 보였다.

주요어: 협업 운송, 비행 로봇, 장애물 회피, 역 기구학, 컨센서스

학 번: 2013-30209